

Universidad de las Ciencias Informáticas

Facultad 6



**Instalador gráfico avanzado de PostgreSQL con facilidad
para la personalización del gestor.**

Trabajo de Diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autor: Carlos Manuel Cruz Nuñez

Tutor: Ing. Yunior Bauta Pentón

Cotutor: Ing. Yudisney Vazquez Ortiz



Junio 2011

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Carlos Manuel Cruz Nuñez

Firma del Autor

Ing. Yudisney Vazquez Ortiz

Firma del Cotutor

Ing. Yunior Bauta Pentón

Firma del Tutor

“Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.”

Albert Einstein



DATOS DE CONTACTO

Tutor: Ing. Yunior Bauta Pentón

Email: ypenton@uci.cu

Cotutor: Ing. Yudisney Vazquez Ortiz

Email: yvazquezo@uci.cu

AGRADECIMIENTOS



A mis padres porque ambos son un ejemplo a seguir, desde pequeño me enseñaron a luchar por lo que se quiere lograr, y hacerlo uno mismo.

A mi princesita Lieny por estar a mi lado todo este tiempo, te quiero mucho.

A mis abuelos Adelina y Rafael porque han sido mis segundos padres, porque son las luces que alumbran mi camino.

A mis hermanos por ser uno de los regalos más grandes que me dio esta vida.

A mis dos tías Lila y Yamilia que son lo que más quiero en este mundo.

A mi suegrita linda Lien y a Papito por tratarme como si fuera un hijo más de ellos, los quiero.

A Mario y a Aylín que cuando tuve alguna dificultad, estuvieron ahí para ayudarme.

A mi cotutora Yudisney y mi tutor Yunion que además de brindarme todo su apoyo y conocimiento se convirtieron en grandes amigos.

A mis compañeros y amigos que estuvieron juntos a mí en estos cinco largos años y que junto a ellos tengo los mejores recuerdos de mi vida.

A los profesores de la Universidad de las Ciencias Informáticas que contribuyeron en mi formación profesional.

A nuestra Revolución y Comandante en Jefe porque sin ellos este sueño no hubiese sido posible.

DEDICATORIA



Este trabajo se lo voy a dedicar en especial a mi mamá Niurka Núñez Reyes que desde pequeño me enseñó a ser independiente, que nada es imposible, que hoy soy el resultado de lo que he querido ser, que cuando pensamos que no se puede más, siempre podemos dar un paso. A ella porque es mi vida y que gracias a ella, hoy soy la persona que quiero ser. A mi tía Yamilia Núñez Reyes porque es una inspiración para mí, porque siempre he estado orgulloso de que sea mi tía. A mi suegrita bella y hermosa y a Papito que se han convertido en unos padres para mí, gracias por toda la ayuda y el apoyo que me han brindado. A mi hermanito Michel y mi primito Alejandro Michel para que esto le sirva como reto en su vida profesional. A mi tía Lila ya que ella es la que siempre me ha estado apoyando en los momentos difíciles y es la que me guía por el buen camino. A Lieny por soportarme todo este año, y por ser tan linda conmigo.

A todas las personas que de una forma u otra me han ayudado en estos cinco años.

RESUMEN

En el año 2002 surge la Universidad de las Ciencias Informáticas (UCI), al calor de la batalla de ideas, con los objetivos de favorecer el desarrollo de la economía cubana y contribuir a la informatización de las instituciones del país. El Centro de Tecnologías de Gestión de Datos (DATEC) es un centro para la producción de software creado en la UCI, el mismo tiene entre sus misiones contribuir a la soberanía tecnológica potenciando el empleo de tecnologías de base de datos de código abierto tomando como base al gestor PostgreSQL.

Surge entonces la presente investigación con el propósito de crear un instalador de una rama de PostgreSQL 9.0 que realice el proceso de instalación partiendo del código fuente y permita seleccionar lenguajes procedurales, opciones del fichero de configuración, la instalación de los módulos de contribución divididos en análisis de datos, administración, monitoreo, desarrollo y seguridad, además que instale documentación del gestor y la herramienta de administración del mismo llamada pgAdmin3. El instalador avanzado de PostgreSQL va a ser utilizado por las empresas u organismos cubanos que requieran de una instalación avanzada del mismo o que necesite funciones que no estén presentes en el gestor. Esta investigación tributa a que se materialice la migración a software libre que debe llevar a cabo el país.

Palabras claves: gestor de base de datos, instalador de PostgreSQL 9.0, módulos de contribución.

TABLA DE CONTENIDOS

| | |
|---|-----|
| DECLARACIÓN DE AUTORÍA..... | I |
| DATOS DE CONTACTO | III |
| AGRADECIMIENTOS..... | IV |
| DEDICATORIA..... | V |
| RESUMEN | VI |
| TABLA DE CONTENIDOS | VII |
| INTRODUCCIÓN..... | a |
| Capítulo I “Fundamentos teóricos” | 5 |
| Introducción..... | 5 |
| 1.1 Sistemas de gestión de bases de datos..... | 5 |
| 1.1.1 Componentes de los sistemas de gestión de bases de datos | 5 |
| 1.2 PostgreSQL 9.0..... | 6 |
| 1.3 Módulos de contribución | 8 |
| Descripción de los módulos de análisis de datos | 8 |
| Descripción de los módulos de monitoreo | 8 |
| Descripción de los módulos de administración..... | 9 |
| Descripción de los módulos de desarrollo | 10 |
| Descripción de los módulos de seguridad | 12 |
| 1.4 Lenguajes procedurales..... | 12 |
| 1.5 Plataforma seleccionada para el desarrollado del instalador | 13 |
| 1.6 Herramientas para la construcción de instaladores en el sistema operativo Linux | 14 |
| IzPack..... | 14 |
| Ubucompiler..... | 15 |

| | |
|---|----|
| BitRock InstallBuilder Professional..... | 15 |
| 1.7 Materiales Empleados para la confección del instalador | 16 |
| PgAdmin3 | 16 |
| GCC | 16 |
| <i>Autotools</i> | 16 |
| Guía para la personalización de PostgreSQL..... | 16 |
| Conclusiones..... | 17 |
| Capítulo II “Propuesta de solución” | 18 |
| Introducción..... | 18 |
| 2.1 Proceso de creación de un instalador con la herramienta BitRock..... | 18 |
| 2.1.1 Detalles del producto..... | 19 |
| 2.1.2 Archivos | 20 |
| 2.1.3 Avanzado..... | 21 |
| 2.1.4 Personalización | 23 |
| 2.2 Script de posinstalación de PostgreSQL..... | 26 |
| 2.3 Construcción del instalador para GNU/Linux y pasos para la configuración básica de la instalación. | 34 |
| Conclusiones..... | 49 |
| Capítulo III “Validación de la propuesta” | 50 |
| Introducción..... | 50 |
| 3.1 Pruebas funcionales..... | 50 |
| 3.1.1 Descripción General | 51 |
| 3.1.2 Condiciones de ejecución | 51 |
| 3.1.3 Sección: Instalación del PostgreSQL 9.0.4 | 54 |

| | |
|---|----|
| 3.1.4 Descripción de las variables..... | 56 |
| 3.2 Creación y funcionamiento de los accesos directos | 58 |
| 3.3 Funcionamiento de la herramienta pgAdmin3..... | 58 |
| 3.4 Funcionamiento de los lenguajes procedurales | 59 |
| 3.5 Funcionamiento de los módulos del <i>contrib</i> | 60 |
| Conclusiones..... | 61 |
| Conclusiones Generales..... | 62 |
| Referencia bibliográfica | 63 |

INTRODUCCIÓN

Las Tecnologías de la Información y las Comunicaciones (TIC) hacen referencia a una amplia gama de servicios, aplicaciones, tecnologías, equipos y programas informáticos, en la revolución digital que ha tenido lugar estos últimos años las mismas tienen enormes repercusiones. Las TIC ofrecen grandes posibilidades para facilitar y acelerar el desarrollo de los países más pobres del mundo.

Cuba no está ajena a este desarrollo tecnológico y creó diferentes empresas para desarrollar software, lo que hizo que en el año 2002 surgiera la Universidad de las Ciencias Informáticas (UCI), al calor de la batalla de ideas, cuya misión es producir software y prestar servicios informáticos a partir de la vinculación estudio-trabajo como modelo de formación, el cual se basa en la integración de los procesos de formación, investigación y producción en torno a una temática, para convertirla en una rama productiva.

Dentro de la UCI se crearon varios centros para la producción de software, uno de ellos es el Centro de Tecnologías de Gestión de Datos (DATEC) que tiene como misiones:

- Contribuir a la soberanía tecnológica potenciando el empleo de tecnologías de bases de datos de código abierto tomando como base a PostgreSQL.
- Proveer soluciones integrales y consultorías relacionadas con la explotación de PostgreSQL.
- Contribuir a la formación de especialistas en tecnologías de bases de datos de código abierto.

DATEC ha seleccionado como Sistema Gestor de Bases de Datos (SGBD) a PostgreSQL por ser reconocido como el gestor de base de datos objeto-relacional de código abierto más potente del mundo, multiusuario, liberado bajo la licencia *Berkeley Software Distribution*¹ (BSD), que soporta gran parte del estándar SQL y ofrece modernas características como consultas complejas, vistas, disparadores, integridad transaccional, control de concurrencia y puede ser extendido por el usuario añadiendo tipos de datos, operadores, funciones agregadas y lenguajes procedurales, que funciona sobre 34 plataformas incluyendo Windows, Linux, FreeBSD, Solaris y Unix; que soporta gran cantidad de lenguajes dentro de su API para el desarrollo de las aplicaciones, como SQL, Java, Perl, Python, C, C++, Ruby y PHP.(1)

¹ A partir de este momento, todas las palabras escritas en idioma Inglés se van a poner en cursiva.

En Cuba actualmente se está promoviendo el empleo de PostgreSQL en aplicaciones empresariales, de manera que se materialice la migración a software de código abierto que debe llevar a cabo el país. Uno de los problemas que más atenta contra la adopción del gestor en las empresas y organismos es lo engorrosa que resulta su instalación cuando se desea personalizar.

La empresa norteamericana *EnterpriseDB* desarrolla instaladores gráficos de PostgreSQL que son llamados por muchos *OneClickInstallers*, por su amigable proceso de instalación mediante clics. Estos tienen como inconveniente que no dan la posibilidad de que el usuario defina los parámetros de configuración del gestor, siendo la instalación más avanzada, la que se realiza partiendo del código fuente del mismo. No permite además la instalación de las contribuciones que pueden ser requeridas por el usuario y no permite la instalación de lenguajes procedurales en el servidor.

Ante esta situación en junio del 2010 el departamento de PostgreSQL del centro DATEC en la UCI, liberó un instalador para la versión 8.4 del gestor que posibilita agregar diferentes módulos de contribución, que son paquetes que se pueden encontrar o colocar en el directorio *contrib* de la distribución de PostgreSQL, pero este instalador tiene una serie de inconvenientes que se listan a continuación:

- No da la posibilidad de que el usuario le defina los parámetros de configuración.
- No instala la herramienta para la administración del gestor.
- No permite instalar lenguajes procedurales de alto nivel que se pueden instalar con el gestor.
- No crea accesos directos para facilitar el trabajo con el gestor.
- Instala la versión 8.4, cuando ya fue liberada la 9.0.

Estos elementos influyen negativamente en la proliferación del uso de PostgreSQL 9.0, incidiendo en el proceso de migración a tecnologías de bases de datos de código abierto del país. Situación por la que surge el siguiente **problema de la investigación** ¿Cómo contribuir al proceso de instalación y personalización de PostgreSQL9.0, facilitando la migración a tecnologías de base de datos de código abierto en empresas, organismos e instituciones cubanas?

Para dar solución al problema antes planteado se define como **objeto de estudio** la creación de instaladores, delimitando el **campo de acción** a la creación de instaladores gráficos para PostgreSQL.

Para dar solución al problema de investigación se define como **objetivo general** desarrollar un instalador gráfico avanzado de PostgreSQL con facilidad para su personalización.

A partir del objetivo general, se derivan los siguientes **objetivos específicos**:

- Realizar un estudio del estado del arte de los instaladores de PostgreSQL.
- Realizar un estudio de los materiales existentes para el desarrollo de los instaladores de PostgreSQL.
- Desarrollar el instalador gráfico avanzado para la plataforma Linux.
- Probar el instalador en la plataforma Linux.

Para cumplir con los objetivos planteados se llevan a cabo las siguientes **tareas de la investigación**:

- Revisión bibliográfica del estado del arte de los instaladores de PostgreSQL.
- Revisión bibliográfica de las herramientas a utilizar para el desarrollo del instalador.
- Identificación de los pasos necesarios para la instalación de PostgreSQL en Linux.
- Definición de los requerimientos para la confección del instalador para PostgreSQL 9.0.
- Desarrollo del instalador gráfico avanzado para PostgreSQL 9.0.
- Prueba del instalador desarrollado.

Resultado esperado: al concluir la presente investigación la comunidad técnica cubana de PostgreSQL debe tener acceso a un instalador gráfico de PostgreSQL 9.0, que a diferencia de los instaladores existentes, permita la instalación de 30 módulos de contribución del gestor y 3 lenguajes procedurales, instale PostgreSQL partiendo del código fuente del mismo, brinde al usuario la posibilidad de la instalación más avanzada y personalizada que se le puede hacer al gestor, instale herramientas de administración y cree accesos directos para facilitar el trabajo con PostgreSQL.

Para el éxito de la presente investigación, se utilizan varios **métodos científicos de investigación** como son: el **método teórico** que posibilita el conocimiento del estado del arte del fenómeno y dentro de este el **método analítico-sintético**, con el que se analiza la necesidad de hacer un instalador de PostgreSQL para Cuba, el estudio de los temas relacionados a él, así como el estudio de las herramientas que se utilizan para darle una respuesta al problema de la investigación.

Para validar la propuesta de solución es necesario el uso de dos **métodos empíricos**, el primero es la **revisión documental** para la comprensión de toda la documentación a trabajar y el segundo es la **observación** para la búsqueda de información y la guía de todo el proceso de investigación.

El presente trabajo está estructurado de la siguiente manera:

Capítulo 1: “Fundamentos teóricos”. Aborda el estudio del estado del arte de los sistemas de gestión de bases de datos, PostgreSQL y PostgreSQL 9.0, se detallan los lenguajes procedurales y los parámetros de configuración, se describen los módulos de contribución y las diferentes herramientas que existen para la construcción de instaladores en sistemas operativos Linux.

Capítulo 2: “Propuesta de solución”. Se describe el proceso de creación de un instalador, se detallan una serie de pasos para la instalación del PostgreSQL en Linux y se describen las secciones del script de posinstalación para la construcción del instalador avanzado con funcionalidades de análisis de datos, administración, monitoreo, desarrollo y seguridad.

Capítulo 3: “Validación de la propuesta”. Se valida la propuesta realizada utilizando como método de prueba, el de caja negra, así como otras pruebas realizadas al instalador para ver cómo reacciona ante determinadas situaciones.

Capítulo I “Fundamentos teóricos”

Introducción

En el presente capítulo se abordan los fundamentos teóricos necesarios para la comprensión del desarrollo de la investigación y se definen conceptos relevantes orientados al tema de investigación. Todo lo anterior tributa a un mejor entendimiento del trabajo actual sobre el desarrollo de un instalador en modo gráfico para PostgreSQL con facilidad para la personalización del gestor partiendo del código fuente.

1.1 Sistemas de gestión de bases de datos

En la actualidad existen diferentes conceptos relacionados con los sistemas de gestión de bases de datos, pero los fundamentales se encuentran entorno a los que se relacionan a continuación:

“Un sistema de gestión de bases de datos es un conjunto de programas que administran y gestionan la información contenida en una base de datos.”(2)

“Un sistema gestor de base de datos es un conjunto de programas que permiten crear y mantener una base de datos, asegurando su integridad, confidencialidad y seguridad.”(3)

“Los sistemas de gestión de bases de datos son un tipo de software que sirven de interfaz entre la base de datos, el usuario y las aplicaciones que la usan.”(4)

De manera general los sistemas de gestión de bases de datos son un conjunto de programas que crean, administran y mantienen toda la información de una base de datos. Los mismos garantizan la integridad, confidencialidad y seguridad de las bases de datos, sirviendo como vía de comunicación entre el usuario y las aplicaciones que utilizan estas bases de datos.

1.1.1 Componentes de los sistemas de gestión de bases de datos

Los SGBD están integrados por diferentes componentes y entre los principales se encuentran(5):

- **Control de autorización:** comprueba que el usuario tiene los permisos necesarios para llevar a cabo la operación que solicita.

- **Procesador de comandos:** una vez que el sistema ha comprobado los permisos del usuario, se pasa el control al procesador de comandos.
- **Control de la integridad:** cuando una operación cambia los datos de la base de datos, este módulo debe comprobar que la operación a realizar satisface todas las restricciones de integridad necesarias.
- **Optimizador de consultas/Planificador:** determina la estrategia óptima para la ejecución de las consultas.
- **Gestor de transacciones:** realiza el procesamiento de las transacciones.
- **Gestor de recuperación:** garantiza que las bases de datos permanezcan en un estado consistente en caso de que se produzca algún fallo.
- **Gestor de búferes:** responsable de transferir los datos entre la memoria principal y los dispositivos de almacenamiento secundario. A este módulo también se le denomina gestor de datos.

1.2 PostgreSQL 9.0

Un ejemplo de sistema de gestión de bases de datos es el PostgreSQL, líder entre los gestores de bases de datos de código abierto. Cuenta con una comunidad global de miles de contribuyentes, usuarios, docenas de compañías y organizaciones. El proyecto PostgreSQL tiene más de 20 años de desarrollo, con un ritmo de desarrollo actual sin precedentes. El conjunto de funcionalidades de PostgreSQL no sólo es comparable a los mejores sistemas gestores de datos propietarios, sino que las superan en características avanzadas, extensibilidad, seguridad y estabilidad.

El PostgreSQL antes del 20 de septiembre de 2010 contaba con diferentes versiones, pero no fue hasta esta fecha que el Grupo Global de Desarrollo de PostgreSQL liberó la versión 9.0 del Sistema de Gestión de Bases de Datos de código abierto PostgreSQL. La nueva versión incorpora características avanzadas en materia de seguridad, soporte de aplicaciones, seguimiento y control, rendimiento y almacenamiento de datos especializados.(6)

A continuación se especifican las principales características:

Replicación y Escalabilidad

- Hot Standby.
- Replicación en flujo.

Administración y Seguridad

- Actualización in-situ integrada (pg_upgrade).
- Autenticación RADIUS.
- Verificación de fortaleza de contraseñas (passwordcheck).
- Órdenes de gestión de permisos más sencillas (GRANT ON ALL y DEFAULT PERMISSIONS).

Diseño de Base de Datos y SQL

- Restricciones de unicidad postergables.
- Triggers condicionales.
- Triggers por columna.
- Ordenamiento en funciones de agregación.
- Nuevas opciones de ventanas deslizantes (ROWS PRECEDING y FOLLOWING).

Procedimientos Almacenados

- Bloques anónimos de código procedural (sentencia DO).
- Procedimientos almacenados en Perl y Python mejorados (incluyendo soporte para Python 3).
- Invocaciones con nombres de parámetros.

Rendimiento y Funciones avanzadas

- Mensajería de eventos mejorada (LISTEN/NOTIFY).
- Soporte a Windows 64-bit.
- Optimización para consultas generados por ORM (eliminación de JOIN).
- Llaves de unicidad para datos no escalares (restricciones por exclusión).
- Expansión del soporte a datos de tipo llave-valor (hstore).
- Planes de EXPLAIN en JSON y XML.

PostgreSQL 9.0 usa la licencia PostgreSQL, similar a la licencia BSD, que sólo requiere que el código fuente mantenga su información de *copyright* y licencia. Esta licencia certificada por OSI es reconocida por ser flexible y amigable para los negocios, puesto que no restringe el uso de PostgreSQL con aplicaciones comerciales y propietarias.(6)

La *EnterpriseDB* no tardó en construir un instalador gráfico para esta versión, pero este instalador tiene una serie de limitaciones en cuanto a configuración del servidor como son:

- No permite la instalación de los diferentes módulos del *contrib*.
- No permite establecer los parámetros del fichero de configuración.
- No permite la instalación de lenguajes procedurales.

1.3 Módulos de contribución

PostgreSQL 9.0 contiene módulos de contribución, los cuales son paquetes que se pueden encontrar o colocar en el directorio *contrib* de la distribución. Incluyen herramientas para migración, utilidades de análisis y características de tipo *plugins* las cuales no son parte del núcleo de PostgreSQL, sobre todo porque son realizadas para un propósito específico o porque son muy experimentales para ser parte del árbol del código fuente principal.

Para utilizar los módulos hay que configurarlos porque cuando se instala desde el código fuente los mismos no lo hacen de manera automática. Los contemplados en el instalador a desarrollar se clasifican en módulos de análisis de datos, monitoreo, administración, desarrollo y seguridad.(5)

Descripción de los módulos de análisis de datos

Los módulos de análisis de datos considerados son los siguientes:

- Módulo **fuzzystrmatch**: proporciona varias funciones para determinar las similitudes y la distancia entre las cadenas. Creado el 27 de febrero del año 2006, sus desarrolladores fueron Bruce Momjian y Joe Conway.
- Módulo **cube**: implementa un tipo de datos para la representación de cubos multidimensionales. Creado el 11 de diciembre del año 2000 por Gene Selkov.

Descripción de los módulos de monitoreo

Los módulos de monitoreo considerados son los siguientes:

- Módulo **pg_buffercache**: proporciona un medio para examinar lo que está sucediendo en el búfer de caché en tiempo real. El módulo ofrece una función `pg_buffercache_pages` que retorna un conjunto de registros. Además de una vista `pg_buffercache` que devuelve la función en un uso más cómodo. Creado el 12 de marzo del año 2005 por Mark Kirkwood.
- Módulo **pg_freespace**: este módulo proporciona un método para examinar el mapa del espacio libre (FSM). Además proporciona una función llamada `pg_freespace` o dos funciones sobrecargadas y las funciones de mostrar el valor registrado en el mapa de espacio libre para una página dada, o para todas las páginas en la relación. Creado el 12 de febrero del año 2006 por Mark Kirkwood y Heikki Linnakangas.
- Módulo **pg_stat_statements**: proporciona un medio para el seguimiento de las estadísticas de la ejecución de todas las sentencias SQL en el servidor. Creado el 4 de enero del año 2009 por Takahiro Itagaki.

Descripción de los módulos de administración

Los módulos de administración considerados son los siguientes:

- Módulo **adminpack**: proporciona una serie de funciones de apoyo utilizadas por la administración y herramientas de gestión para funcionalidades adicionales, como la gestión remota de los archivos de registro del servidor. En el PostgreSQL 8.2 es la primera versión donde se incluye al `adminpack` como módulo en el *contrib* el 30 de mayo del 2006 y su desarrollador fue Andreas Pflug.
- Módulo **oid2name**: es una utilidad que ayuda a los administradores a examinar la estructura de archivos utilizados por PostgreSQL. Requiere de un servidor de base de datos ejecutándose con los catálogos de sistemas no corruptos. Por tanto, es de uso limitado para recuperarse de situaciones de bases de datos de corrupción catastrófica. Creado el 24 de enero del año 2001 por B. Palmer.
- Módulo **pgbench**: es un programa para ejecutar las pruebas de referencia sobre el PostgreSQL. Es ejecutado en la misma secuencia de los comandos SQL, posiblemente en bases de datos de múltiples sesiones concurrentes y después calcula la tasa promedio de operaciones (operaciones por segundos). Creado el 15 de enero del año 2000 por Tatsuo Ishii, autor de `pgPool`.

- Módulo **lo**: proporciona apoyo para la gestión de objetos grandes. Creado el 16 de junio del año 1998 por Peter Mount.
- Módulo **pg_standby**: apoya la creación de un servidor suplente de bases de datos. Está diseñado para ser un programa de producción listo, así como una plantilla personalizable en caso de necesitar modificaciones específicas. Creado el 8 de febrero del año 2007 y su creador fue Simon Riggs.
- Módulo **pgrowlocks**: proporciona una función para mostrar información del bloqueo de registros de una tabla especificada. Creado el 23 de abril del año 2006 por Tatsuo Ishii.
- Módulo **pgstattuple**: proporciona diversas funciones para obtener estadísticas a nivel de tupla. Creado el día 1 del mes de octubre del año 2001 por Tatsuo Ishii y Satoshi Nagayasu.
- Módulo **pageinspect**: proporciona funciones que le permiten inspeccionar el contenido de las páginas de bases de datos en un nivel bajo, que es útil para propósitos de depuración. Su fecha de creación fue el 17 de mayo del año 2007 por Heikki Linnakangas.

Descripción de los módulos de desarrollo

Los módulos de desarrollo considerados son los siguientes:

- Módulo **auto_explain**: proporciona un medio para registrar automáticamente los planes de ejecución de sentencias lentas. Esto es especialmente útil para la optimización de las consultas en las aplicaciones de gran tamaño. Creado el 19 de noviembre del año 2008 por Takahiro Itagaki.
- Módulo **hstore**: implementa un tipo de datos para almacenar conjuntos de pares (clave, valor), dentro de un único campo de datos PostgreSQL; útil en diversos escenarios tales como las filas con muchos atributos que rara vez son examinados o en datos semi-estructurados. Creado el 5 de septiembre del 2006 por Oleg Bartunov y Teodor Sigaev.
- Módulo **ltree**: implementa un tipo de datos ltree para la representación de las etiquetas de los datos almacenados en un árbol de estructura jerárquica. Sus desarrolladores fueron Teodor Sigaev y Oleg Bartunov y fue creado el 30 de julio del año 2002.
- Módulo **pg_trgm**: proporciona funciones y operadores para determinar las similitudes de texto basado en la coincidencia trigramas, así como clases de operadores de índices que soportan la

búsqueda rápida para cadenas similares. Creado el 3 de mayo del 2004 por Teodor Sigaev y Oleg Bartunov.

- Módulo **seg**: implementa un tipo de datos para la representación de segmentos de línea, o intervalos de punto flotante; puede representar la incertidumbre en las variables de intervalos. Creado el 11 de diciembre del 2000 por Gene Selkov.
- Módulo **tablefunc**: incluye diversas funciones que devuelven varias filas. Estas funciones son útiles tanto en su propio derecho como en ejemplos para escribir funciones en el lenguaje C que retornen múltiples filas. Creado el 30 de julio del año 2002 por Joe Conway.
- Módulo **uuidosp**: proporciona funciones para generar identificadores únicos universales (UUID) usando uno de varios algoritmos estándares. También tiene funciones especiales para producir ciertas constantes UUID y fue creado el 21 de abril del año 2007 por Peter Eisentraut.
- Módulo **citext**: proporciona un tipo de cadena de caracteres en mayúsculas y minúsculas. Creado el 29 de julio del 2008 por David E. Wheeler.
- Módulo **dict_xsyn** (diccionario de sinónimos extendido): es un ejemplo de un diccionario plantilla añadido para la búsqueda de texto completo, este tipo de diccionario sustituye palabras con los grupos de sus sinónimos, haciendo posible la búsqueda de una palabra con alguno de sus sinónimos. Desarrollado el 15 de octubre del 2007 por Sergey Karpov.
- Módulo **spi**: proporciona varios ejemplos viables para usar SPI y disparadores. Mientras que estas funciones son de algún valor en su propio derecho, además son más útiles como ejemplos para modificar sus propios fines. Las funciones son suficientemente generales como para ser utilizadas en cualquier tabla, pero se tienen que especificar los nombres de las tablas y campos. Desarrollado el 11 de septiembre del año 1997 por Teodor Sigaev y Oleg Bartunov.
- Módulo **test_parser**: es un ejemplo de un analizador de encargo para la búsqueda de texto completo. No hace nada especialmente útil, pero puede servir como punto de partida para el desarrollo de su propio intérprete. Desarrollado el 15 de octubre del 2007 por Sergey Karpov.

- Módulo **btree_gin**: proporciona una muestra de una clase de operador Gin10 que implementa el comportamiento equivalente de BTree (que son estructuras de árbol que se encuentran comúnmente en las implementaciones de bases de datos y sistemas de archivos) para diferentes tipos de datos. Creado el 25 de marzo del 2009 por Teodor Sigaev y Oleg Bartunov.
- Módulo **btree_gist**: ofrece una muestra de clase de operador Gist11 que implementa Btree equivalente al comportamiento para diferentes tipos de datos. Desarrollado el 28 de mayo del 2004 por Teodor Sigaev, Oleg Bartunov y Janko Richter.
- Módulo **earthdistance**: ofrece dos enfoques diferentes para calcular las distancias de gran círculo en la superficie de la Tierra. En este módulo, se supone que la tierra es perfectamente esférica. Creado el 16 de junio del año 1998 por Bruno Wolff III y Hal Snyder.
- Módulo **isn**: proporciona un tipo de datos isn para el producto internacional de numeración de diferentes estándares. Proporciona un operador de comparación estándar, además de indexaciones btree y hash, para todos esos tipos de datos. Creado el 9 de septiembre del año 2006 por Germán Méndez Bravo.

Descripción de los módulos de seguridad

Los módulos de seguridad considerados son los siguientes:

- Módulo **chkpss**: implementa un chkpass, que es un tipo de datos que está diseñado para almacenar contraseñas encriptadas, cada contraseña se convierte automáticamente en forma cifrada a la entrada y siempre se almacena de esa manera. Desarrollado el 3 de mayo del 2001 por D'Arcy J.M. Cain.
- Módulo **pgcrypto**: implementa funciones criptográficas para PostgreSQL. Creado el 31 de octubre del 2000 por Marko Kreen.

1.4 Lenguajes procedurales

Los lenguajes procedurales ofrecen a los programadores conceptos que se traducen de forma natural al modelo de los ordenadores. A partir del lanzamiento de la versión 6.3 de PostgreSQL, se soporta la definición de lenguajes procedurales.

En el caso de una función o procedimiento definido en un lenguaje procedural, la base de datos no tiene un conocimiento implícito sobre cómo interpretar el código fuente de las funciones. El manejador en sí es una función de un lenguaje de programación compilada en forma de objeto compartido, y cargado cuando es necesario.(7)

Existen varios lenguajes procedurales como son: PL/Java, PL/R, PL/Ruby, PL/pgSQL, PL/Perl, PL/Python, PL/Tcl y los seleccionados para el instalador gráfico son:

- **PL/Perl:** Lenguaje procedural imperativo que permite más control que el álgebra relacional del SQL. Las funciones creadas en PL/Perl pueden tener las características del lenguaje Perl.
- **PL/Python:** Lenguaje procedural que permite utilizar dentro de PostgreSQL funciones escritas en python.
- **PL/Tcl:** Lenguaje procedural para el gestor de bases de datos PostgreSQL que permite el uso de Tcl para la creación de funciones y procedimientos desencadenados por eventos.

Aunque los lenguajes descritos anteriormente son los seleccionados para el instalador gráfico porque son los más usados se hace necesario destacar que el PL/pgSQL lo incluye implícitamente el PostgreSQL 9.0.

1.5 Plataforma seleccionada para el desarrollo del instalador

Linux es la denominación de un sistema operativo y el nombre de un núcleo. El código fuente está disponible públicamente y cualquier persona puede libremente usarlo, modificarlo y redistribuirlo. El término Linux se utiliza para describir al sistema operativo tipo Unix que utiliza filosofías y metodologías libres y que está constituido por la combinación del núcleo Linux con las bibliotecas y herramientas del proyecto GNU. Tiene dos características que lo diferencian de los otros sistemas operativos, la primera es que es libre y la segunda es que el sistema viene acompañado del código fuente. Se distribuye bajo la licencia GNU *Public License*, dentro de sus principales funciones se encuentran que es un sistema multitarea y multiusuario. Fue creado por Linus Torvalds en la universidad de Helsinki en Finlandia y su primera versión oficial, versión 0.02 fue anunciada el 5 de octubre de 1991. (8)

1.6 Herramientas para la construcción de instaladores en el sistema operativo Linux

Los programas en los sistemas GNU/Linux vienen generalmente empaquetados en archivos DEB para las distribuciones basadas en Debian, archivos RPM entre otros. Existen herramientas para la creación de instaladores en GNU/Linux como son:

IzPack

IzPack es una herramienta para la construcción de instaladores, de código abierto, es esencialmente una aplicación Java, es multiplataforma, crea instaladores que funcionan en cualquier sistema operativo que tenga instalado el *Java Runtime Environment*. La versión 3.3.0 se distribuye bajo licencia GPL y entre sus características se destacan:

- Configuración de la instalación en formato XML.
- Optimiza la compresión de los ficheros JAR.
- Integración con Ant.
- Fácilmente extensible.
- Creación automática de un desinstalador.
- Disponible en 16 idiomas.

IzPack proporciona un conjunto de utilidades:

- Un lanzador de aplicaciones nativas que se pueden utilizar para verificar primero que hay un entorno de ejecución de Java (JRE) instalado y a continuación iniciar los instaladores
- Una herramienta que ajusta instaladores dentro de paquetes ejecutables autoextraíbles Windows.
- Una herramienta que envuelve a los instaladores dentro de paquetes de aplicaciones Mac OS X.

IzPack comenzó en 2001 gracias a Julien Ponge. Su última versión estable disponible es la 4.3.3 con licencia *Apache License 2.0* y en idioma inglés. Tiene como inconveniente que es una aplicación Java.(9)(10)

Ubucompiler

Es un programa de código abierto para crear instaladores. Es un programa realizado con Gambas, un lenguaje de programación derivado de BASIC, que permite generar paquetes .deb a partir del código fuente de los programas, compatibles con Debian y sus distribuciones derivadas: Ubuntu, Linux Mint, Super OS y viene en versiones para los idiomas inglés y español.(11)

BitRock InstallBuilder Professional

BitRock InstallBuilder es una herramienta de desarrollo multiplataforma para la creación de instaladores para equipos de escritorio y software de servidor. (12)

Entre sus características se destacan:

- Genera paquetes RPM y Debian.
- Soporta los lenguajes Java, PHP, Perl, Python, Ruby, C++.
- Ofrece actualización automática para que siempre se ejecute la mejor versión disponible.
- Crea rápidamente instaladores profesionales para Linux, Windows, Mac OS X, Solaris y otras plataformas.

Los instaladores creados con esta herramienta tienen las siguientes características:

- No requieren de un tiempo de ejecución en Java.
- Son completamente autónomos.
- No tienen dependencias externas.
- Realizan una copia de seguridad de todos los archivos sobrescritos durante la instalación por lo que en caso de error se puede recuperar.
- Están optimizados en tamaño y velocidad y no requieren de un proceso de extracción.
- Se ejecutan automáticamente en modo de interfaz gráfica y modo texto.

Para la construcción del instalador se selecciona BitRock InstallBuilder por las ventajas antes mencionadas que este ofrece respecto a las otras estudiadas.

1.7 Materiales Empleados para la confección del instalador

Los materiales seleccionados para la construcción del instalador son:

PgAdmin3

Es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia *Open Source*. PgAdmin3 está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. La herramienta también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados y cuenta con soporte para el motor de replicación Slony-I.(13)

GCC

Otro de los materiales utilizados para la construcción del instalador es el *GNU Compiler Collection*, es uno de los proyectos insignes de la *Free Software Foundation*, constituye el pilar angular para el desarrollo del gestor al contar con el estándar C89, sobre el cual está implementado PostgreSQL. Va a ser el compilador seleccionado para llevar a cabo la inclusión de los módulos en la instalación del gestor.(14)

Autotools

Conjunto de herramientas para la distribución y estandarización del código del gestor, que van a posibilitar compilar el código fuente del mismo de la forma clásica

/configure, make y make install.

Guía para la personalización de PostgreSQL

Procedimiento para la compilación de las fuentes del gestor y de cada uno de los módulos de análisis de datos, monitoreo, administración, desarrollo y seguridad. Fue desarrollada en la UCI por Aldo Cristiá Álvarez y Yeni Morgado Sánchez.

Conclusiones

En el presente capítulo fue realizado un estudio sobre conceptos fundamentales que consolidan un mejor entendimiento del trabajo, así como las herramientas y la plataforma para la construcción de instaladores. Se selecciona el BitRock como herramienta de empaquetado de software para la creación de instaladores, el GCC como compilador para llevar a cabo la inclusión de cada uno de los módulos, las *Autotools* para compilar el código fuente de PostgreSQL y la Guía de Personalización del mismo. Además fueron analizados los módulos que pueden ser añadidos al gestor.

Capítulo II “Propuesta de solución”

Introducción

Este capítulo aborda la construcción del instalador avanzado cubano para el sistema Linux, se detallan una serie de pasos necesarios para la configuración básica de la instalación, se muestra el proceso de creación de un instalador utilizando la herramienta BitRock y las secciones del script de posinstalación que va a ser utilizado para la instalación del gestor.

2.1 Proceso de creación de un instalador con la herramienta BitRock

La herramienta BitRock cuenta con una interfaz sencilla y amigable, su pantalla principal, ver Figura 1, donde proporciona una pequeña bienvenida y brinda la posibilidad de crear un nuevo proyecto o abrir uno existente.

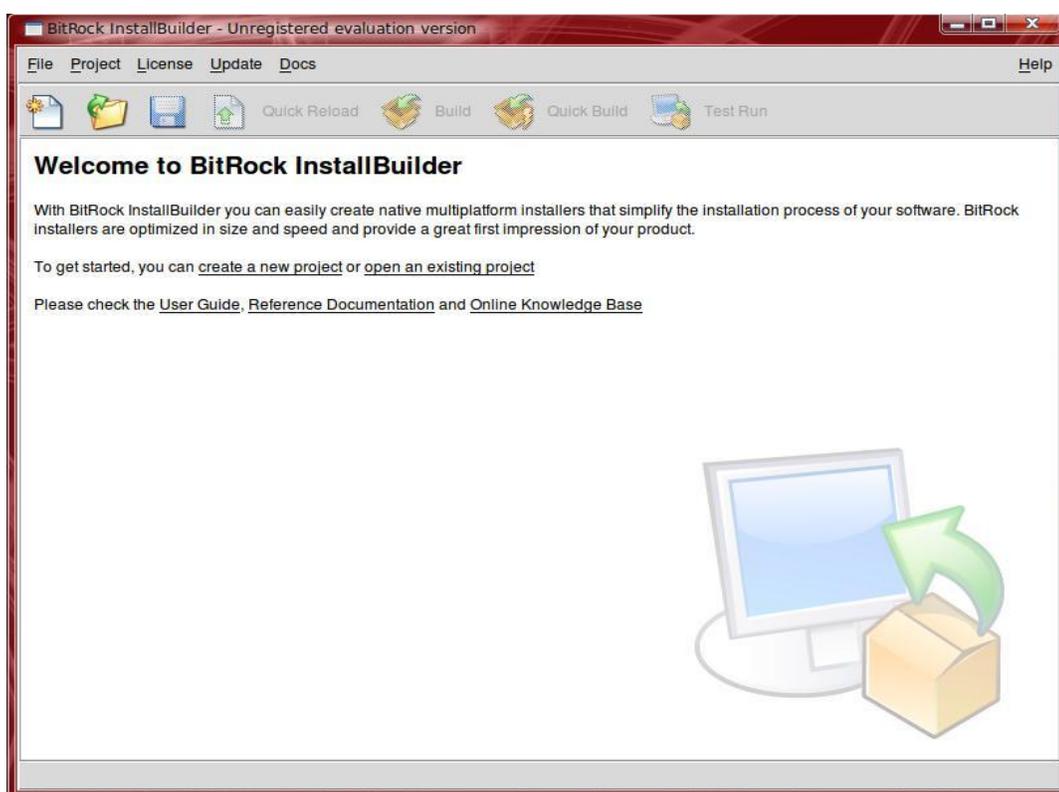


Figura 1. Bienvenido a BitRock InstallBuilder

2.1.1 Detalles del producto

Una vez creado el nuevo proyecto la herramienta cuenta con cinco opciones fundamentales para la confección del instalador, en la opción Detalles del Producto (del inglés *Product Details*), ver Figura 2, se introducen una serie de datos para la creación del instalador como son:

- Nombre de la Compañía (*Vendor name*): se introduce el nombre de la compañía que hace el producto, para este proyecto el nombre es Comunidad Técnica Cubana de PostgreSQL.
- Nombre del Producto (*Product Name*): se debe mostrar el nombre del producto que va a ser empaquetado, para este caso el nombre es PostgreSQL.
- Nombre del Instalador Binario (*Product Filename*): es el nombre con el cual se guarda el instalador.
- Número de la Versión (*Version Number*): versión del producto, para este proyecto es la versión 9.0.4.
- Fichero Léeme (*Readme File*): en este caso se debe seleccionar la dirección de un fichero de texto que contiene la información para ser leída por el usuario al terminar el proceso de instalación.
- Fichero de Licencia (*License File*): se debe seleccionar un fichero de texto con los acuerdos de licencia que se le establezca al producto.
- Fichero HTML de Licencia (*HTML License File*): es la misma licencia pero en formato html.

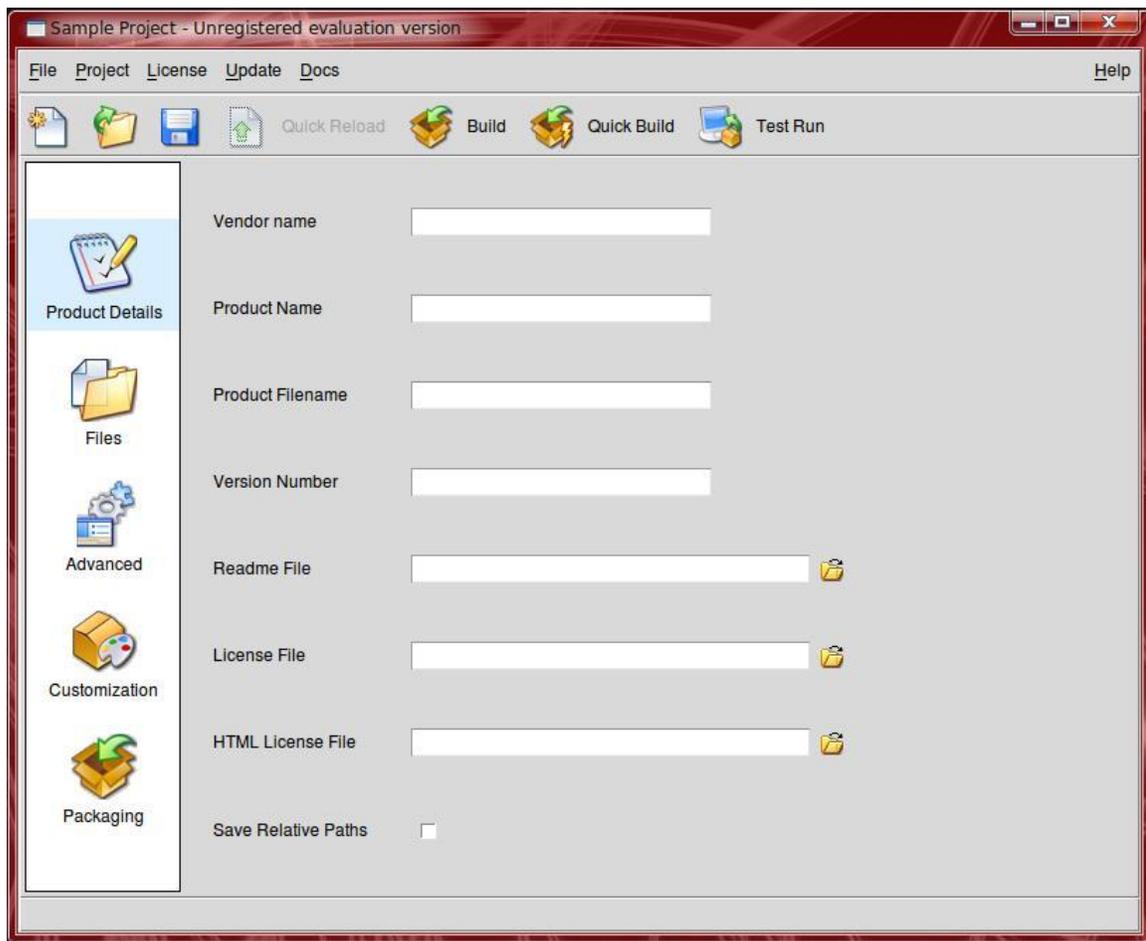


Figura 2. Product Details

2.1.2 Archivos

En la opción Archivos (del inglés *Files*), ver Figura 3, se encuentran los archivos de programas para las diferentes plataformas (Windows, Linux y Mac OS X), una vez seleccionada la plataforma en la que va a ser creado el producto, se cargan cuatro archivos fundamentales, sin importar el orden los archivos son:

- Script de posinstalación: es el que hace posible el proceso de instalación del gestor.
- Script de predesinstalación: se encarga de remover PostgreSQL del sistema.
- Fuente de una rama de PostgreSQL 9.0.

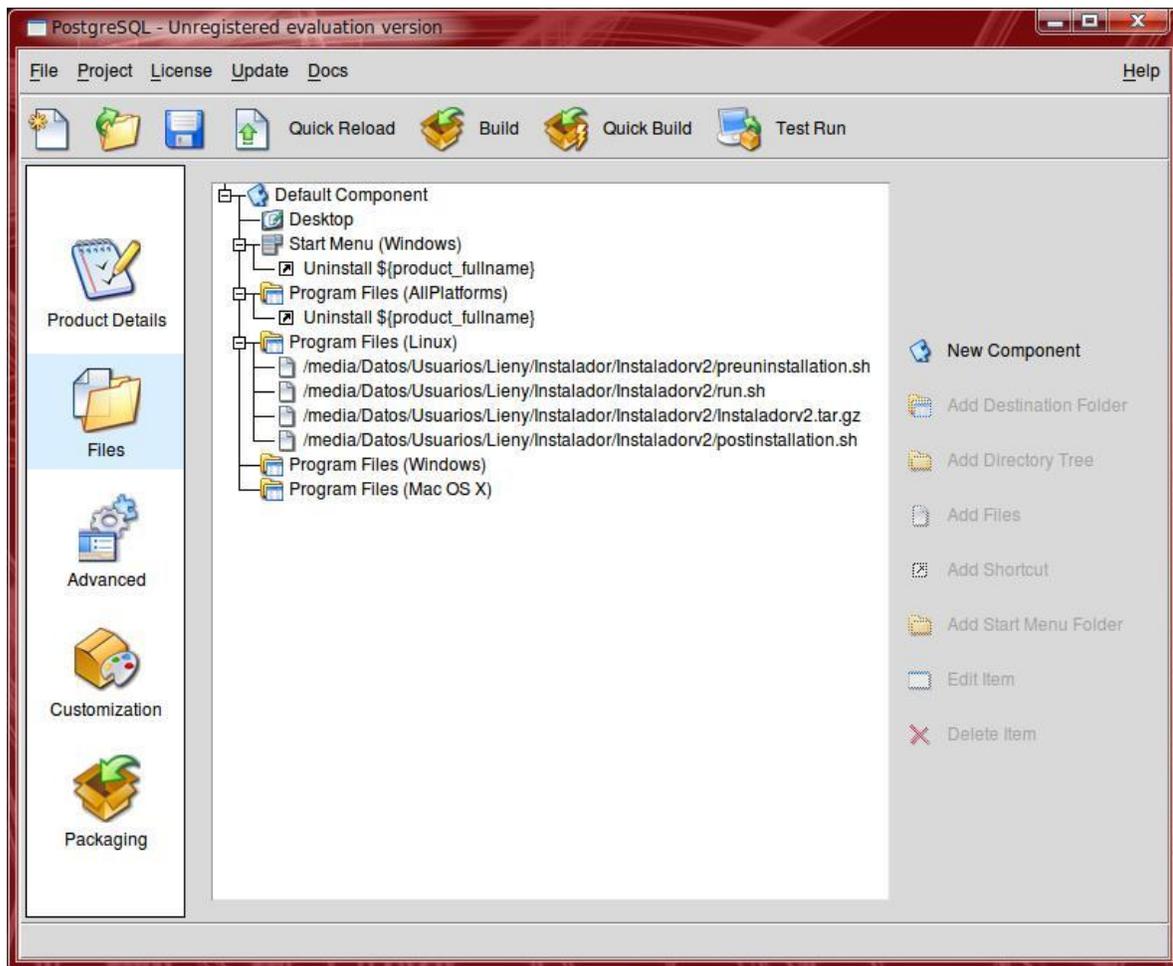


Figura 3.Files

2.1.3 Avanzado

En la opción Avanzado (del inglés *Advanced*), ver Figura 4, es donde se crean todas las ventanas del instalador, BitRock da la posibilidad de crear ventanas de tipo:

- *Boolean*: utilizadas para la recepción de valores de tipo booleano en forma de selección, puede ser como checkbox y radiobuttons.
- *Choice*: pide al usuario que seleccione un valor entre una lista predefinida.
- *Directory*: pide al usuario que introduzca un directorio.
- *File*: pide al usuario que introduzca un archivo.
- *Info*: muestra un texto de solo lectura.

- *Label*: muestra el parámetro de la etiqueta pasado a una cadena de texto de sólo lectura.
- *License*: muestra una licencia.
- *Link*: muestra un enlace como una etiqueta o un botón, que ejecuta una acción cuando se hace clic sobre él.
- *Parameter Group*: permite crear páginas personalizadas.
- *Password*: permite al usuario entrar una contraseña.
- *String*: permite solicitar una cadena de texto por parte del usuario (acepta todas las opciones comunes).

Las ventanas del instalador son:

- Directorio de Instalación: directorio donde es instalado el Gestor.
- Directorio de Datos: directorio donde se guardan las bases de datos y los ficheros de configuración.
- Contraseña: contraseña para el superusuario postgres.
- Puerto: puerto por donde acepta conexiones el servidor.
- Lenguajes Procedurales: lenguajes procedurales a instalar.
- Módulos de Análisis de Datos: módulos de análisis de datos a instalar.
- Módulos de Monitoreo: módulos de monitoreo que se desean instalar.
- Módulos de Administración: módulos de administración a instalar.
- Módulos de Desarrollo: módulos de desarrollo a instalar.
- Módulos de Seguridad: módulos de seguridad a instalar.

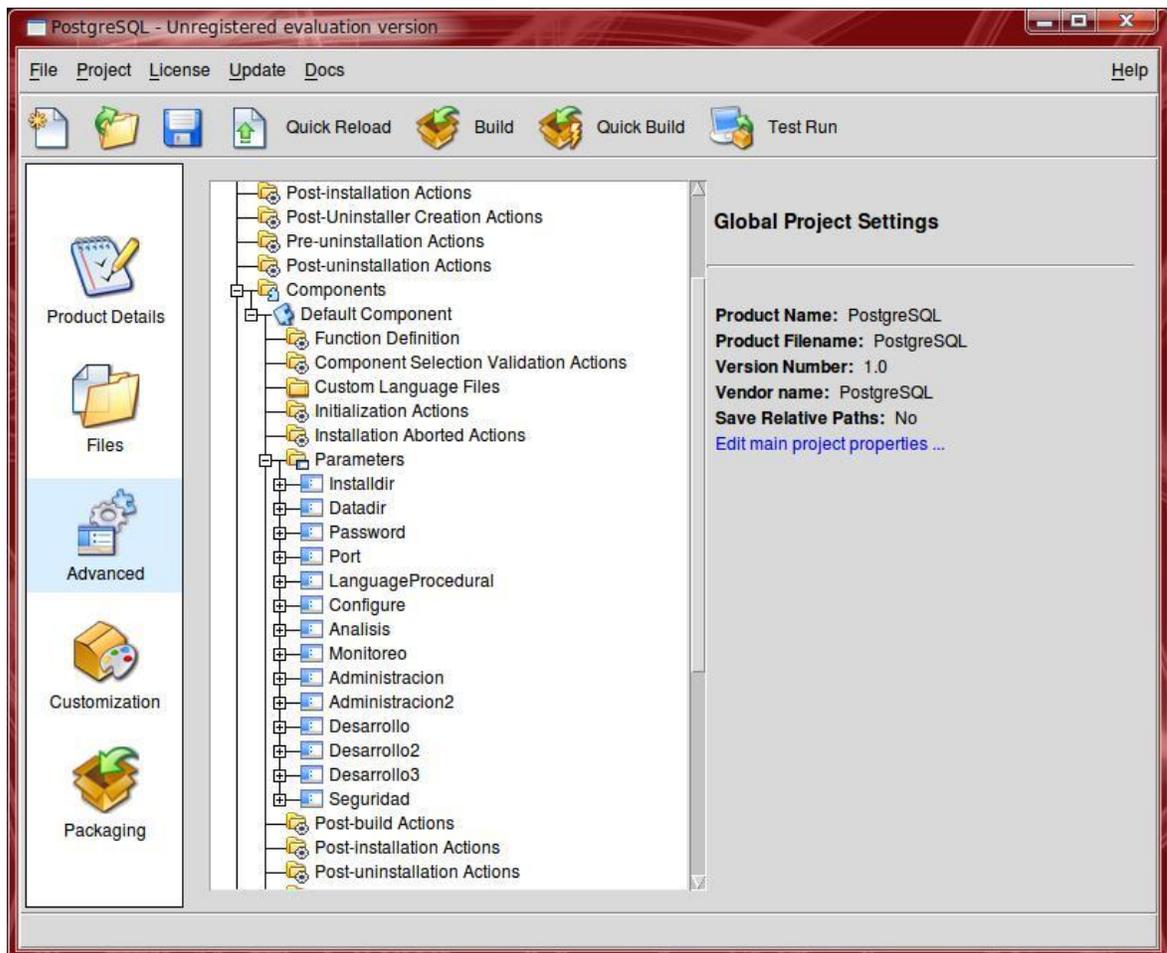


Figura 4.Advanced

2.1.4 Personalización

En la opción de Personalización (del inglés *Customization*), tiene varias secciones, en la sección *User Interface*, ver Figura 5, se introducen una serie de datos para la creación del instalador, los mismos son:

- Imagen de Logo (*Logo Image*): en este caso se debe seleccionar la dirección de una imagen que se muestra en las ventanas del instalador.
- Imagen del Lado Izquierdo (*Left Side Image*): se debe seleccionar una imagen que se muestra en el lado izquierdo de las ventanas del instalador.
- Imagen de Fondo (*Splash Image*): se debe seleccionar una imagen que se muestra en el fondo de las ventanas del instalador.

- Selección del Lenguaje de Instalación (*Default Installation Language*): se selecciona el lenguaje para llevar a cabo el proceso de instalación.

En la sección de *Scripts*, ver Figura 6, se introducen una serie de datos muy importantes para la creación del instalador, tales como:

- Script de Posinstalación (*Post-Install Script*): se ejecuta una vez terminado el desempaquetado de los ficheros.
- Argumentos del Script de Posinstalación (*Post-Install Script Arguments*): se escriben todas las variables del script de posinstalación en el mismo orden en que aparecen de la siguiente manera `${nombre_variable}`.
- Script de Predesinstalación (*Pre-Uninstallation Script*): se ejecuta cuando se desinstala PostgreSQL.
- Argumentos del Script de Predesinstalación (*Pre-Uninstallation Script Arguments*): se escriben las variables del script de predesinstalación.

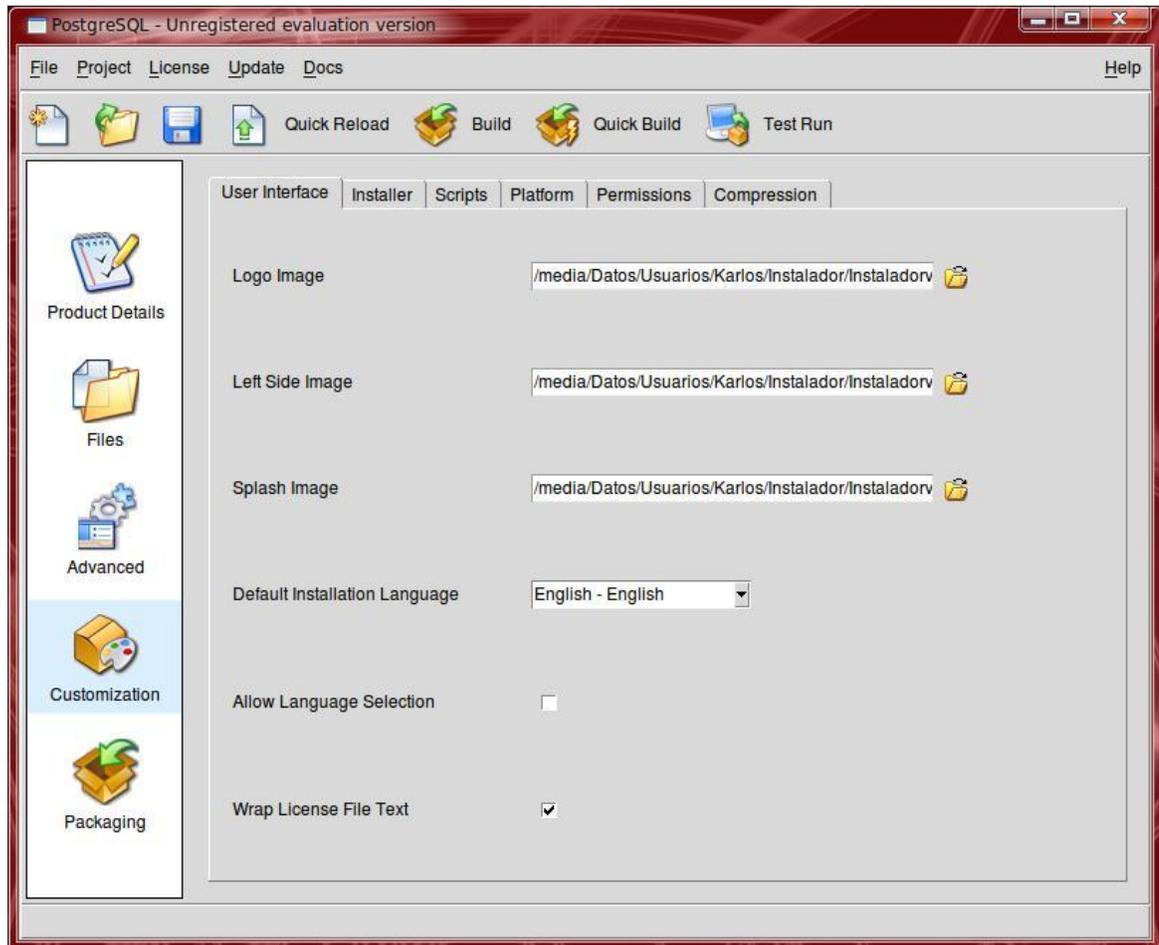


Figura 5. Customization, Sección User Interface.

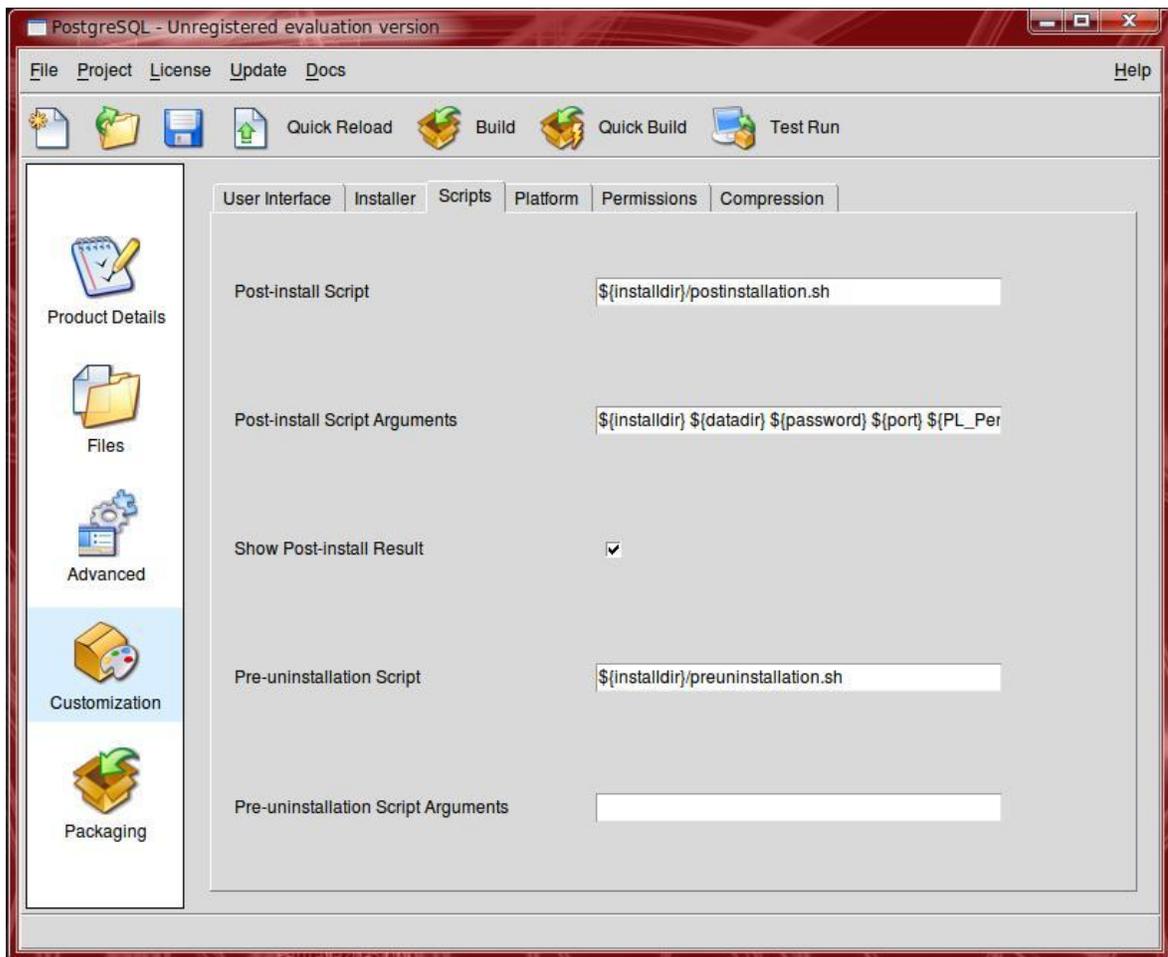


Figura 6. Customization, Sección Scripts.

2.2 Script de posinstalación de PostgreSQL

El instalador avanzado va a contar con varias secciones las mismas son:

1- Recepción de las variables del BitRock: para la captura de las variables en la herramienta BitRock se debe poner en el mismo orden como se declaran en el script de posinstalación, para ello en la parte de personalización en la herramienta existe un campo que se llama argumentos del script de posinstalación, a continuación se muestra una imagen de las variables declaradas en el script de posinstalación.

```
installdir=${1}
datadir=${2}
password=${3}
port=${4}
#recibir lenguajes procedurales a instalar.
pl_perl=${5}
pl_python=${6}
pl_tcl=${7}
#recibir parámetros del configure a instalar.
enable_nls=${8}
with_openssl=${9}
with_pam=${10}
#recibir módulos a instalar de análisis de datos
fuzzystrmatch=${11}
pl_r=${12}
#recibir módulos a instalar de monitoreo
pg_buffercache=${13}
pg_freespacemap=${14}
pg_stat_statements=${15}
#recibir módulos a instalar de administración
adminpack=${16}
oid2name=${17}
pg_bench=${18}
lo=${19}
pg_standby=${20}
pg_rowslocks=${21}
pgstattuple=${22}
pageinspect=${23}
#recibir módulos a instalar de desarrollo
auto_explain=${24}
hstore=${25}
ltree=${26}
pg_trgm=${27}
seg=${28}
table_func=${29}
uuid_osp=${30}
citext=${31}
dict_xsyn=${32}
spi=${33}
test_parser=${34}
btree_gin=${35}
btree_gist=${36}
earthdistance=${37}
isn=${38}
cube=${39}
#recibir módulos a instalar de seguridad
chkpass=${40}
pgcrypto=${41}
```

2- Instalación de las librerías necesarias con los siguientes comandos:

```
apt-get -y build-essential bison flex libc6 libcomerr2 libkrb5-3 tclx8.4-dev  
apt-get -y install libreadline5-dev libpam0g libpq5 libssl10.9.8 libpam0g-dev  
apt-get -y install zlib1g-dev libssl-dev openssl tzdata ssl-cert libwxgtk2.8-0
```

3- Extracción de la fuente (.tar.gz): para lograr una mayor rapidez en el proceso de instalación se tiene compactado el código fuente del PostgreSQL 9.0.4 y en el script de posinstalación se extrae mediante el comando:

```
tar -xvzf codigofuente.tar.gz
```

4- Instalación de los lenguajes procedurales y los parámetros del fichero configure.

```
cadena="./configure --prefix=$installdir"
if [ $pl_perl = 1 ];
then
cadena=$cadena" --with-perl"
echo Instalado Perl
fi
if [ $pl_python = 1 ];
then
cadena=$cadena" --with-python"
echo Instalado Python
fi
if [ $pl_tcl = 1 ];
then
cadena=$cadena" --with-tcl"
echo Instalado tcl
fi
if [ $enable_nls = 1 ];
then
cadena=$cadena" --enable-nls"
echo Instalado enablenls
fi
if [ $with_openssl = 1 ];
then
cadena=$cadena" --with-openssl"
echo Instalado openssl
fi
if [ $with_pam = 1 ];
then
cadena=$cadena" --with-pam"
echo Instalado pam
fi
echo $cadena
$cadena
make
make install-docs
make install
```

5- Agregando usuario Postgres y estableciendo los locales.

```
mkdir $datadir
adduser -gecos Postgres --home $installdir --disabled-password postgres
chown -R postgres $datadir
chown -R postgres $installdir

su postgres -c "$installdir/bin/initdb --locale=${locale} -D $datadir"
su postgres -c "$installdir/bin/postgres -D $datadir&"
```

6- Estableciendo contraseña para el superusuario postgres

```
passw="'$password'"
supostgres -c "psql -d template1 -c "ALTER USER postgres WITH PASSWORD
'${passw}'";"
```

7- Creando fichero PostgreSQL en directorio /etc/init.d

```
echo "Creando fichero postgresql en directorio /etc/init.d para iniciar
postgresql..."

echo "#!/bin/sh" >/etc/init.d/postgresql

echo 'echo "PostgreSQL start/stop"' >>/etc/init.d/postgresql

echo "POSTGRESQL_OWNER=postgres" >>/etc/init.d/postgresql

echo "POSTGRESQL_HOME=/usr/local/pgsql" >>/etc/init.d/postgresql

echo 'case "$1" in' >>/etc/init.d/postgresql

echo "'start')'" >>/etc/init.d/postgresql

echo 'echo -n "Starting the PostgreSQL Server... "' >>/etc/init.d/postgresql
```

```
echo 'POSTMASTER="$POSTGRESQL_HOME/bin/postmaster \-i \-D $POSTGRESQL_HOME/data >>
$POSTGRESQL_HOME/data/bitacora.btc 2>&1&"' >>/etc/init.d/postgresqld

echo 'su - $POSTGRESQL_OWNER -c "$POSTMASTER"' >>/etc/init.d/postgresqld

echo "echo" >>/etc/init.d/postgresqld

echo ";;" >>/etc/init.d/postgresqld

echo "'stop')'" >>/etc/init.d/postgresqld

echo 'echo -n "Shutting down PostgreSQL Server... "' >>/etc/init.d/postgresqld

echo 'KILLPOSTMASTER="kill \-INT `head -1 $POSTGRESQL_HOME/data/postmaster.pid`"'
>>/etc/init.d/postgresqld

echo 'su - $POSTGRESQL_OWNER -c "$KILLPOSTMASTER"' >>/etc/init.d/postgresqld

echo "echo" >>/etc/init.d/postgresqld

echo ";;" >>/etc/init.d/postgresqld

echo "'restart')'" >>/etc/init.d/postgresqld

echo 'echo -n "Restarting PostgreSQL Server... "' >>/etc/init.d/postgresqld

echo "\$0 stop" >>/etc/init.d/postgresqld

echo "\$0 start" >>/etc/init.d/postgresqld

echo "echo" >>/etc/init.d/postgresqld

echo ";;" >>/etc/init.d/postgresqld

echo "*)" >>/etc/init.d/postgresqld

echo 'echo "Usage: postgresqld [ start | stop | restart ]"'
>>/etc/init.d/postgresqld

echo "exit 1" >>/etc/init.d/postgresqld

echo "esac" >>/etc/init.d/postgresqld

echo "exit 0" >>/etc/init.d/postgresqld

chmod 700 /etc/init.d/postgresqld

/etc/init.d/postgresqld start

update-rc.d postgresqld defaults

echo "Creado scrip de arranque de postgres llamado postgresqld"

/etc/init.d/postgresqld "stop"

/etc/init.d/postgresqld "start"
```

8- Configuración del puerto de escucha de Postgres

```
cat $datadir/postgresql.conf | sed -e "s/#port = 5432/port = $port/" >
/tmp/postgresql.conf
mv /tmp/postgresql.conf $datadir/postgresql.conf
```

9- Variables de entorno

```
echo "export PATH=$installdir/bin:$PATH">>/etc/bash.bashrc
echo "export PGDATA=$datadir">>/etc/bash.bashrc
echo "export PGDATABASE=postgres">>/etc/bash.bashrc
echo "export PGUSER=postgres">>/etc/bash.bashrc
echo "export PGPORT=$port">>/etc/bash.bashrc
echo "export MANPATH=$MANPATH:$installdir/share/man">>/etc/bash.bashrc
source /etc/bash.bashrc
echo "creadas las variables de entorno en /etc/bash.bashrc"
```

10- Agregando lenguaje PL/Perl, PL/Tcl y PL/Python.

```
if [ $plperl = 1 ];
then
su postgres -c "psql -d template1 -c "CREATE LANGUAGE plperl"";
if [ $pltcl = 1 ];
then
su postgres -c "psql -d template1 -c "CREATE LANGUAGE pltcl"";
if [ $plpython = 1 ];
then
su postgres -c "psql -d template1 -c "CREATE LANGUAGE plpythonu"";
```

11- Instalación de los módulos de contribución

```
supostgres -c "/usr/local/pgsql/bin/psql -f /usr/local/src/postgresql-9.0/contrib/nom_módulo/ nom_módulo.sql -d template1"

make

make install

#*****INSTALANDO FUZZYSTRMATCH*****

if [ $fuzzystrmatch = 1 ];
then
# instalar FUZZYSTRMATCH
module=contrib/fuzzystrmatch
cd ${installdir}/${folder}/${module}
make
make install
supostgres -c "psql -f
${installdir}/${folder}/contrib/fuzzystrmatch/fuzzystrmatch.sql -d template1"
echo "***** INSTALADO fuzzystrmatch *****"
```

12- Remover Postgres

```
/etc/init.d/postgresql stop
#rm -R /usr/local/src/postgresql-9.0.4
rm -R /usr/local/pgsql/
rm /etc/init.d/postgresql
rm -R /usr/lib/postgresql/
rm -R /var/run/postgresql
userdel -f postgres
```

2.3 Construcción del instalador para GNU/Linux y pasos para la configuración básica de la instalación.

Las principales pantallas para la configuración básica durante la instalación del PostgreSQL Empresarial Cubano son mostradas en las figuras siguientes, ver Figura 1, en esta figura se le brinda al usuario una pequeña bienvenida a la instalación de PostgreSQL, que incluye el logotipo de PostgreSQL, el centro por el que ha sido empaquetado el producto, así como tres botones para la navegación de la aplicación, donde aparece en esta primera imagen el botón Atrás deshabilitado por ser esta la primera interfaz del producto, el botón Adelante para avanzar un paso en la instalación y el Cancelar para cancelar el proceso de instalación.

Al igual que los instaladores ya existentes empaquetados por la *EnterpriseDB* el instalador que se propone tiene una interfaz muy sugerente y fácil de utilizar. A diferencia de los existentes el instalador brinda la funcionalidad de instalar lenguajes procedurales de alto nivel en el servidor, de instalar diferentes módulos de contribución y de opciones avanzadas.



Figura 1: Bienvenido al instalador de PostgreSQL 9.0.4.

Paso 1. Dar clic en el botón Adelante para continuar el proceso de instalación, mostrándose la Figura 2, que especifica el directorio donde va a ser instalado el PostgreSQL con todo lo necesario para su correcto funcionamiento, dígame: librerías, módulos del *contrib*, documentación. En dicho directorio una vez terminada la instalación se limpia el directorio quedando los directorios de PostgreSQL y el desinstalador para que pueda ser desinstalado con facilidad.



Figura 2: Directorio donde va a ser instalado el Gestor

Paso 2. Dar clic en el botón Adelante para continuar el proceso de instalación, mostrándose la Figura 3, donde se selecciona el directorio en que van a estar las bases de datos y los ficheros de configuración, separando la carpeta data de la carpeta de instalación de postgres para que en caso de algún fallo en el sistema operativo no perder los datos.

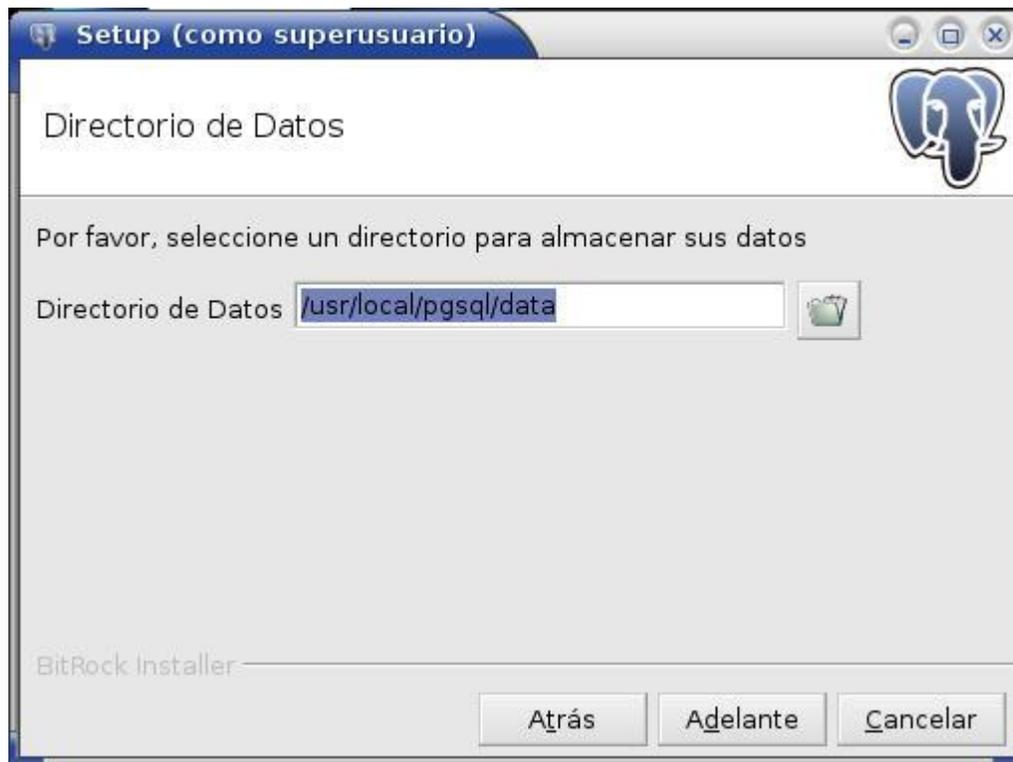


Figura 3: Directorio donde se va a guardar las bases de datos y los ficheros de configuración.

Paso 3. Dar clic en el botón Adelante para continuar el proceso de instalación, mostrándose la Figura 4, donde se introduce la contraseña que tiene el superusuario postgres del servidor PostgreSQL para administrar las bases de datos y sus objetos.

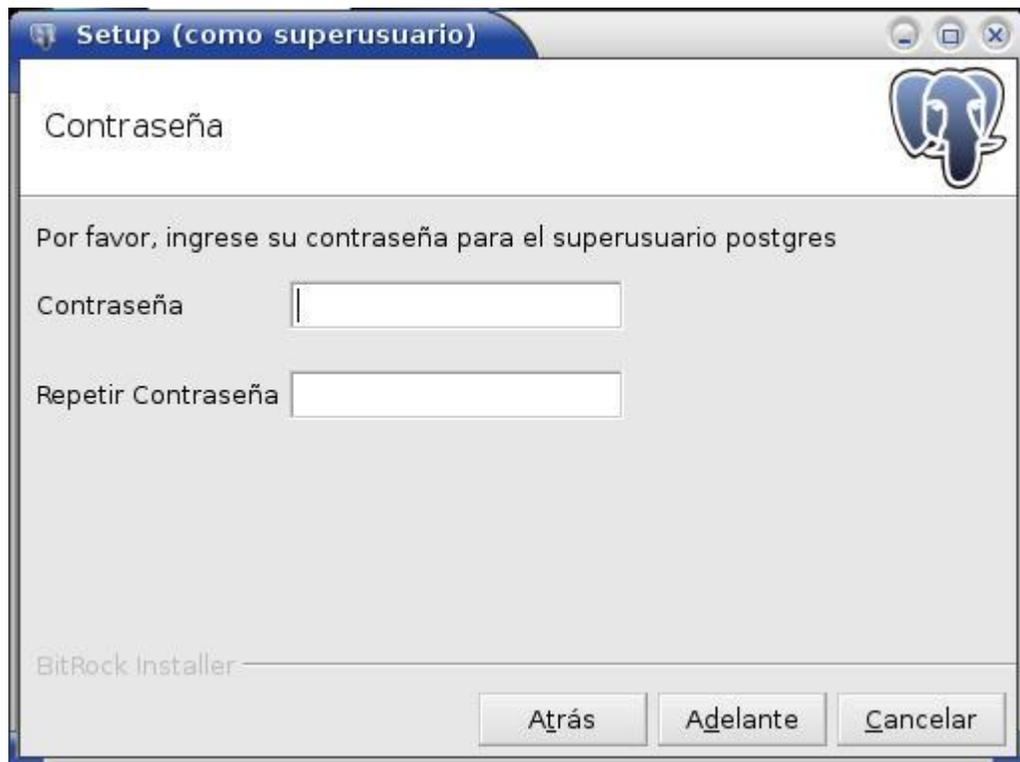


Figura 4: Definición de la contraseña del superusuario postgres.

Paso 4. Dar clic en el botón Adelante para continuar el proceso de instalación, mostrándose en la Figura 5, donde el usuario puede definir el puerto por el que acepta conexiones el servidor.

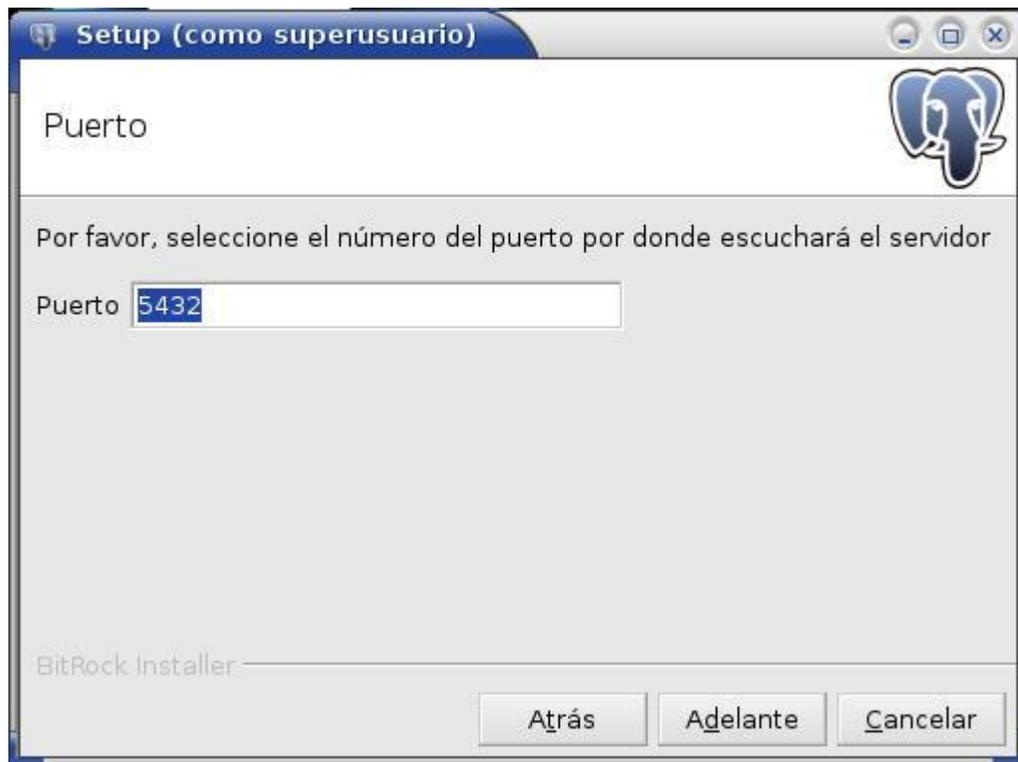


Figura 5: Puerto de escucha del servidor

Paso 5. Dar clic en el botón Adelante para continuar el proceso de instalación, mostrándose la Figura 6, donde se selecciona(n) el/los lenguaje(s) procedural(es) que se desea(n) instalar en las bases de datos.



Figura 6: Selección de los lenguajes procedurales a instalar.

Paso 6. Dar clic en el botón Adelante para continuar el proceso de instalación, mostrándose la Figura 7, donde se seleccionan los módulos de análisis de datos que se desean instalar, para hacer uso de nuevas funciones en las bases de datos.



Figura 7: Módulos de análisis de datos incluidos.

Paso 7. Dar clic en el botón Adelante para continuar el proceso de instalación, mostrándose la Figura 8, donde se seleccionan los módulos de monitoreo que se desean instalar, para hacer uso de otras funciones en las bases de datos.



Figura 8: Módulos de monitoreo incluidos.

Paso 8. Dar clic en el botón Adelante para continuar el proceso de instalación, mostrándose las Figuras 9 y 10 donde se seleccionan los módulos de administración que se desean instalar, para hacer uso de nuevas funcionalidades en las bases de datos.

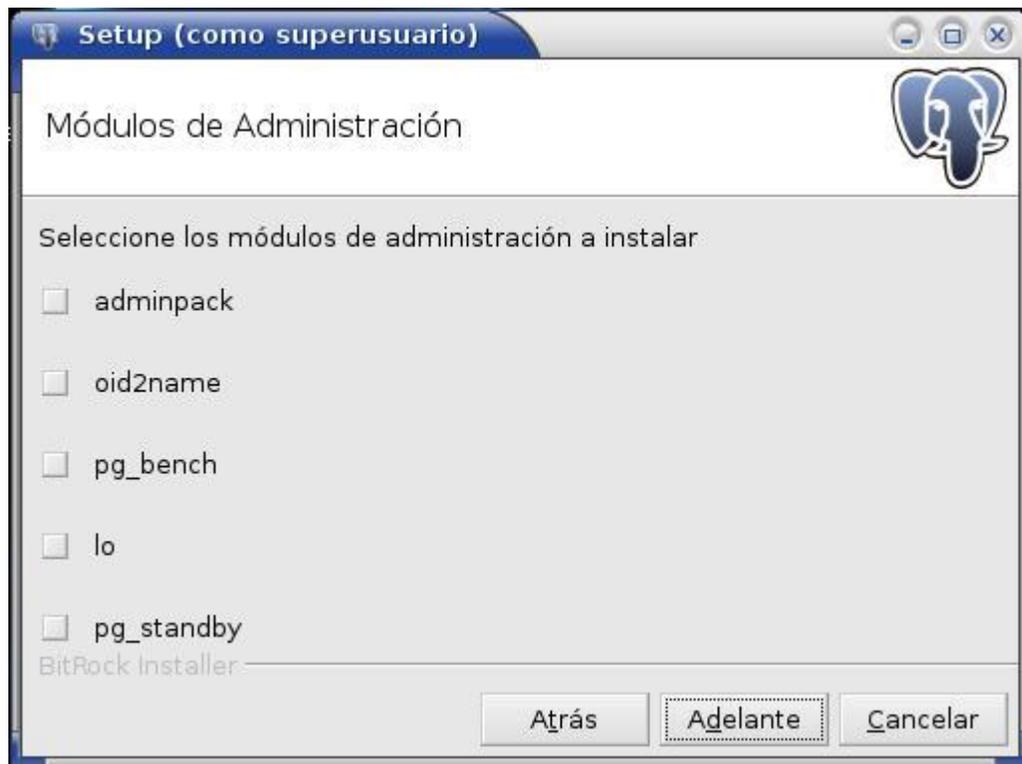


Figura 9: Módulos de administración incluidos.



Figura 10: Módulos de administración incluidos.

Paso 9. Dar clic en el botón Adelante para continuar el proceso de instalación, mostrándose las Figuras 11, 12 y 13 donde se seleccionan los módulos de desarrollo que se desean instalar, para utilizar nuevas funciones en las bases de datos.

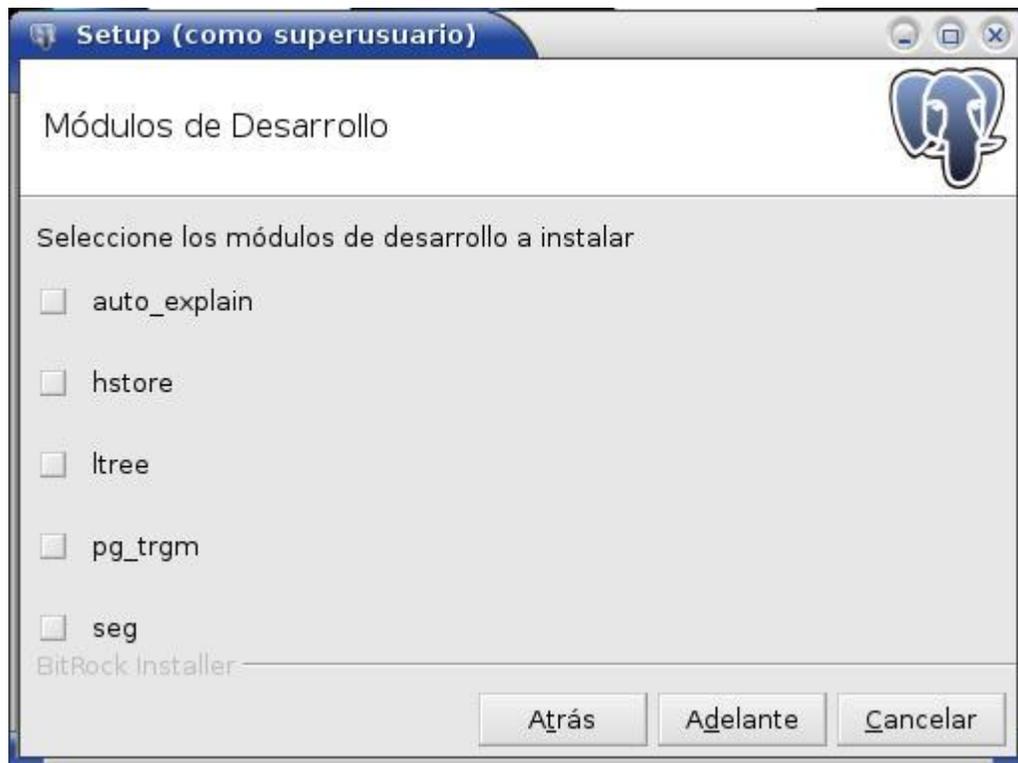


Figura 11: Módulos de desarrollo incluidos.

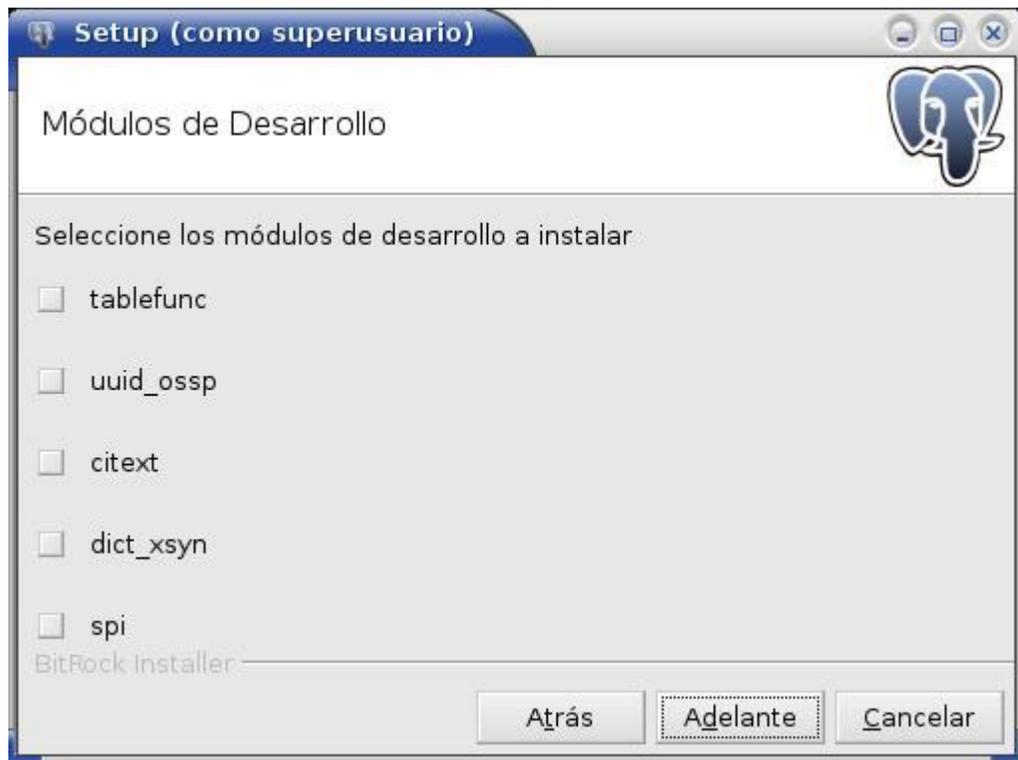


Figura 12: Módulos de desarrollo incluidos.

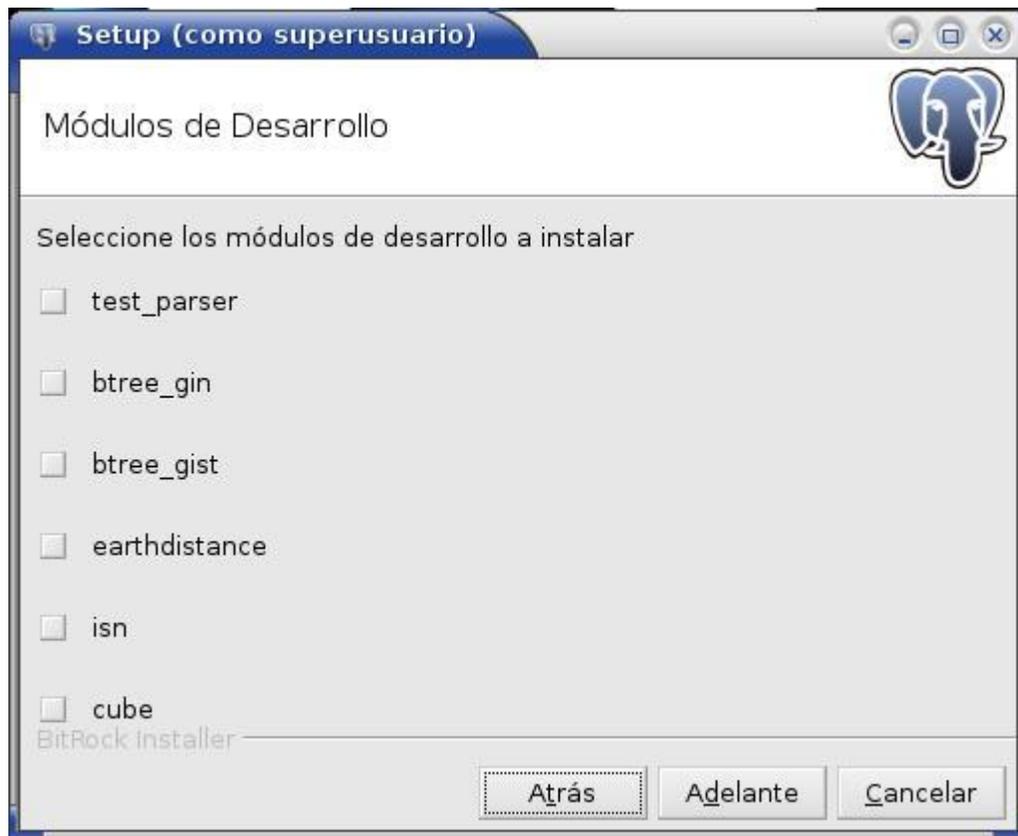


Figura 13: Módulos de desarrollo incluidos.

Paso 10. Dar clic en el botón Adelante para continuar el proceso de instalación, mostrándose la Figura 14, donde se seleccionan los módulos de seguridad que se desean instalar, para hacer uso de nuevas funciones en las bases de datos.



Figura 14: Módulos de seguridad incluidos.

Paso 11. Dar clic en el botón Adelante para continuar el proceso de instalación, mostrándose la Figura 15, donde se muestra una ventana que notifica al usuario que el gestor está listo para instalarse en la computadora.



Figura 15: Instalación lista para llevarse a cabo.

Paso 12. Dar clic en el botón Adelante para continuar el proceso de instalación, mostrándose la Figura 16, para que comience la instalación de PostgreSQL personalizado en su ordenador.



Figura 16: Proceso de instalación en curso.

Paso 13. Dar clic en el botón Terminar para finalizar el proceso de instalación, ver Figura 17, una vez dado el clic en el botón se muestra un archivo el cual brinda algunos datos sobre los realizadores y créditos.



Figura 17: Fin del proceso de instalación.

Conclusiones

Durante el desarrollo del presente capítulo se realizó el proceso de creación de un instalador con la herramienta seleccionada, se plasmaron las secciones del script de posinstalación y se realizó una guía con los pasos necesarios para la configuración básica del instalador.

Capítulo III “Validación de la propuesta”

Introducción

En este capítulo se muestran los resultados de la validación del instalador avanzado con funcionalidades de análisis de datos, monitoreo, administración, desarrollo y seguridad. Para la realización de la evaluación técnica de la propuesta descrita en el capítulo anterior, se va a probar el correcto funcionamiento del instalador y los lenguajes procedurales incluidos.

3.1 Pruebas funcionales

La prueba del software es un elemento crítico para la garantía de la calidad del software. Su objetivo es diseñar pruebas que saquen a la luz diferentes clases de errores con la menor cantidad de tiempo y espacio. De esta manera, se contribuye a preservar las características de calidad que distinguen al software en desarrollo.

En la realización de las pruebas se definen los niveles de prueba a ser ejecutados en concordancia con las características del proyecto. En cada uno de los niveles, la estrategia debe definir qué tipos de pruebas se van a realizar así como métodos y las técnicas a implementar para cada uno de ellos.

Los niveles se clasifican en cuanto a dos criterios, el primero de ellos define a qué elemento se le realiza la prueba, estos son Nivel de Unidad o Prueba Unitaria, Nivel o Prueba de Integración y Nivel o Prueba de Sistema. El otro criterio define quién es el encargado de hacer las pruebas, dentro de estos se enumeran Pruebas de Desarrollador, Pruebas Independientes y Pruebas de Aceptación.

Para validar la propuesta realizada en el capítulo anterior se va a utilizar el Nivel o Prueba de Sistema, son las pruebas que se hacen cuando el software está funcionando como un todo. Es la actividad de prueba dirigida a verificar el programa final, después que todos los componentes de software y hardware han sido integrados. Los métodos convencionales de prueba de caja negra, pueden usarse para dirigir estas pruebas.

Las pruebas de caja negra o también conocidas como pruebas funcionales, pruebas de caja opaca, pruebas de entrada/salida, pruebas inducidas por los datos, son las que no toman en cuenta el código, el

que lo prueba no sabe cómo está estructurado por dentro el programa, solo necesita saber cuáles pueden ser las posibles entradas sin necesidad de entender cómo se deben obtener las salidas, donde se trata de encontrar errores en la interfaz mientras se está usando. (15)

Como herramienta en las pruebas de caja negra se utiliza la determinación de los casos de prueba de sistema, a partir de los requisitos establecidos. Se realiza un caso de prueba por cada requisito funcional.

Un caso de prueba incluye:

- Una descripción general donde se realiza un resumen del requisito funcional.
- Las condiciones de ejecución que debe tener el sistema para probar la funcionalidad del caso de prueba.
- Las secciones de prueba que contienen escenarios y para cada uno de ellos descripción, variables que intervienen, respuesta del sistema y flujo central.
- Descripción de las variables, para tener conocimiento de los valores válidos que admiten las mismas y si estas pueden o no tener un valor.

Para darles valores a las variables se tienen en cuenta 3 valores:

- V: valor válido.
- I: valor inválido.
- NA: no es necesario proporcionar un valor del dato en este caso.

A continuación se muestra el caso de prueba que permite garantizar la calidad de instalación del PostgreSQL 9.0.4.

3.1.1 Descripción General

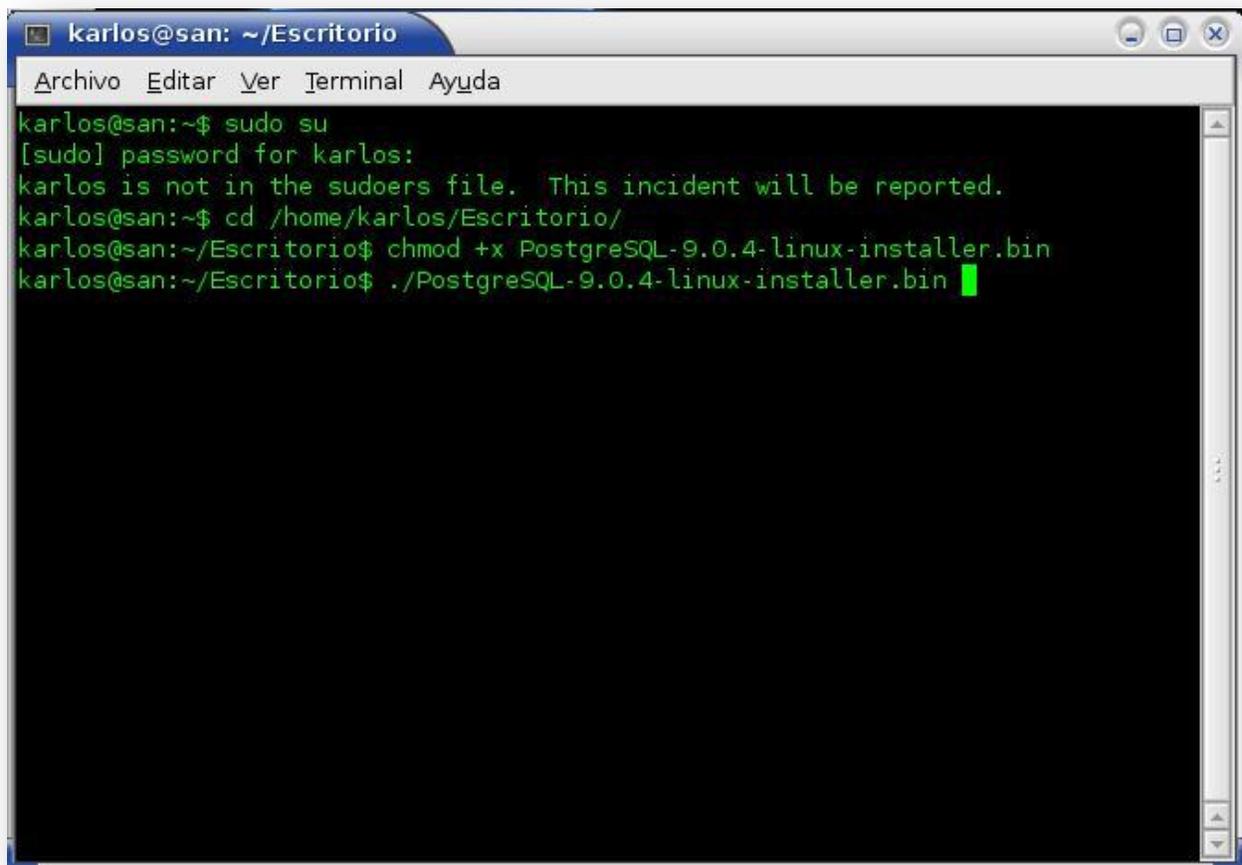
El caso de prueba inicia cuando el usuario procede a instalar el PostgreSQL 9.0.4, el mismo transita por una serie de ventanas que le permiten la instalación de lenguajes procedurales, opciones del fichero de configuración y los módulos de análisis de datos, monitoreo, administración, desarrollo y seguridad.

3.1.2 Condiciones de ejecución

- El usuario tiene que tener abierto un terminal y estar registrado como root.

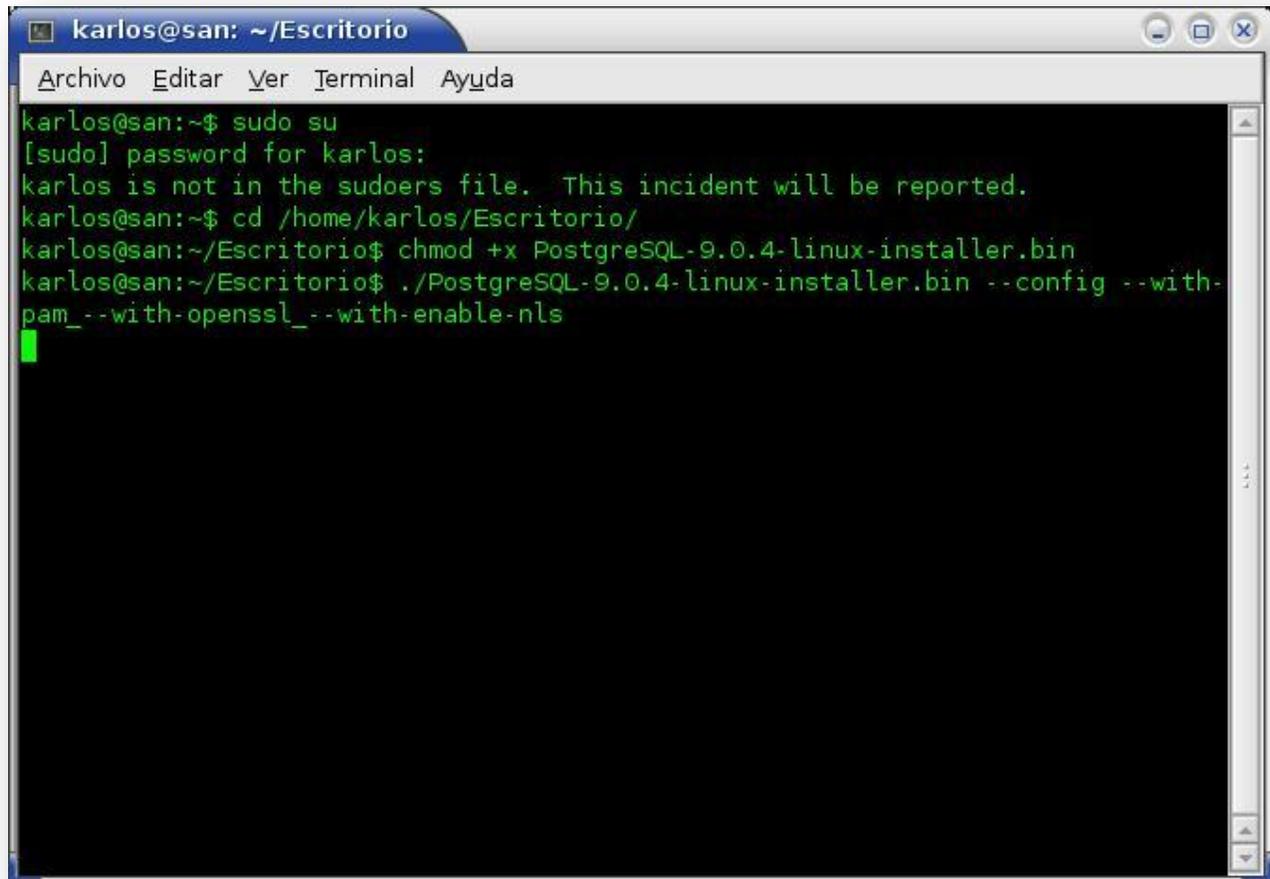
- El usuario debe darle permiso de ejecución al instalador.
- Estar conectado a los repositorios para la descarga efectiva de paquetes necesarios para la compilación de los módulos, lenguajes procedurales y opciones del fichero de configuración.

El instalador cuenta con dos formas de ejecución, la primera realiza una instalación que se le puede agregar lenguajes procedurales y módulos de análisis de datos, monitoreo, administración, desarrollo y seguridad, y la segunda forma de instalación cuenta con las opciones de la primera y además con las opciones avanzadas del fichero de configuración. A continuación se muestra dos imágenes, ver Figuras 1 y 2.



```
karlos@san: ~/Escritorio
Archivo  Editar  Ver  Terminal  Ayuda
karlos@san:~$ sudo su
[sudo] password for karlos:
karlos is not in the sudoers file. This incident will be reported.
karlos@san:~$ cd /home/karlos/Escritorio/
karlos@san:~/Escritorio$ chmod +x PostgreSQL-9.0.4-linux-installer.bin
karlos@san:~/Escritorio$ ./PostgreSQL-9.0.4-linux-installer.bin
```

Figura 1. Primera forma de ejecución del instalador.



```
karlos@san: ~/Escritorio
Archivo Editar Ver Terminal Ayuda
karlos@san:~$ sudo su
[sudo] password for karlos:
karlos is not in the sudoers file. This incident will be reported.
karlos@san:~$ cd /home/karlos/Escritorio/
karlos@san:~/Escritorio$ chmod +x PostgreSQL-9.0.4-linux-installer.bin
karlos@san:~/Escritorio$ ./PostgreSQL-9.0.4-linux-installer.bin --config --with-
pam --with-openssl --with-enable-nls
```

Figura 2. Segunda forma de ejecución del instalador.

3.1.3 Sección: Instalación del PostgreSQL 9.0.4

| Escenario | Descripción | Variabl e 1 | Varia ble 2 | Varia ble 3 | Varia ble 4 | Varia ble 5 | Varia ble 6 | Varia ble 7 | Varia ble 8 | Varia ble 9 | Varia ble 10 | Varia ble 11 | Respuesta del sistema | Flujo central |
|--|--|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|--------------|--------------|---|--|
| EC 1.1 Instalación exitosa de PostgreSQL 9.0.4. | Permite instalar exitosamente el PostgreSQL 9.0.4. | V | V | V | V | V | NA | NA | NA | NA | NA | NA | Muestra información acerca de los módulos instalados correctamente. Crea el acceso directo de la herramienta de administración del gestor (pgAdmin3) en el escritorio. Crea documentación de PostgreSQL 9.0.4 en el escritorio. | 1. Abrir un terminal y autenticarse como root. 2. Dar permisos de ejecución al PostgreSQL-9.0.4-linux-installer.bin. 3. Ejecutar el PostgreSQL-9.0.4-linux-installer.bin. 4. Presionar Enter. 5. Dar clic en el botón Adelante. 6. Seleccionar un directorio válido de instalación. |
| EC 1.2 Validar datos de entrada. | Permite validar la contraseña del superusuario postgres. | NA | NA | I | I | NA | NA | NA | NA | NA | NA | NA | Muestra un mensaje de error "Tienes que especificar un valor". | 7. Seleccionar un directorio válido de almacenamiento de datos. 8. Entrar contraseña del superusuario postgres. 9. Reescribir contraseña del superusuario postgres. 10. Especificar puerto válido. 11. Puede seleccionar el/los lenguaje(s) procedural(es) deseado(s). 12. Dar clic en el botón Adelante. 13. Puede seleccionar el/los módulo(s) de análisis de datos. 14. Dar clic en el botón Adelante. 15. Puede seleccionar el/los módulo(s) |
| | | | | Vacío | Vacío | | | | | | | | | |
| | | NA | NA | V | I | NA | NA | NA | NA | NA | NA | NA | Muestra un mensaje de error "Las contraseñas no coinciden". | |
| | | | | postgr es | Vacío | | | | | | | | | |
| NA | NA | NA | NA | I | V | NA | NA | NA | NA | NA | NA | NA | Muestra un mensaje de error "Tienes que especificar un valor". | |
| | | | | Vacío | postgr es | | | | | | | | | |
| NA | NA | NA | NA | V | V | NA | NA | NA | NA | NA | NA | NA | Muestra un mensaje de error "Las contraseñas no coinciden". | |
| | | | | postgr es | pepe | | | | | | | | | |

CAPÍTULO III INVALIDACIÓN DE LA PROPUESTA

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|----|----|----|----|----|----|----|--|--|
| | | | | | | | | | | | | | | | <p>de monitoreo.</p> <p>16. Dar clic en el botón Adelante.</p> <p>17. Puede seleccionar el/los módulo(s) de administración.</p> <p>18. Dar clic en el botón Adelante.</p> <p>19. Puede seleccionar el/los módulo(s) de desarrollo.</p> <p>20. Dar clic en el botón Adelante.</p> <p>21. Puede seleccionar el/los módulo(s) de seguridad.</p> <p>22. Dar clic en el botón Adelante.</p> <p>23. Esperar hasta que se complete la instalación.</p> <p>24. Dar clic en el botón OK.</p> <p>25. Dar clic en botón Terminar para finalizar la instalación.</p> <p>26. Dar clic en el botón OK.</p> |
| EC 1.3 Instalación incorrecta de PostgreSQL 9.0.4. | No se realiza la instalación de PostgreSQL 9.0.4. | V | V | V | V | V | NA | <p>Muestra un mensaje de error "Permiso denegado."</p> <p>1. Abrir un terminal y no autenticarse como root.</p> <p>2. No dar permisos de ejecución al PostgreSQL-9.0.4-linux-installer.bin.</p> <p>3. Ejecutar el PostgreSQL-9.0.4-linux-installer.bin.</p> <p>4. Presionar Enter.</p> <p>5. Dar clic en el botón Adelante.</p> <p>6. Seleccionar un directorio válido de instalación.</p> <p>7. Seleccionar un directorio válido de</p> | |

| | | | | |
|--------------------|-----------------------------|-------------------------|----|---------------------------|
| Variable 8 | Módulo(s) de monitoreo | Casilla de verificación | Sí | Puede estar seleccionada. |
| Variable 9 | Módulo(s) de administración | Casilla de verificación | Sí | Puede estar seleccionada. |
| Variable 10 | Módulo(s) de desarrollo | Casilla de verificación | Sí | Puede estar seleccionada. |
| Variable 11 | Módulo(s) de seguridad | Campo de contraseña | Sí | Puede estar seleccionada. |

3.2 Creación y funcionamiento de los accesos directos

Se realizaron pruebas para verificar que los accesos directos fueron creados por el instalador y funcionan correctamente. Instalando la documentación de PostgreSQL 9.0.4, la herramienta pgAdmin3 y su documentación, ver Figura 19.



Figura 19: Accesos directos creados en el escritorio

3.3 Funcionamiento de la herramienta pgAdmin3

Se realizaron pruebas para verificar que la herramienta pgAdmin3 funciona correctamente y el éxito de la instalación del gestor.

En la Figura 20 que aparece a continuación se muestra una imagen de la herramienta en su proceso de conexión con el PostgreSQL 9.0.4.

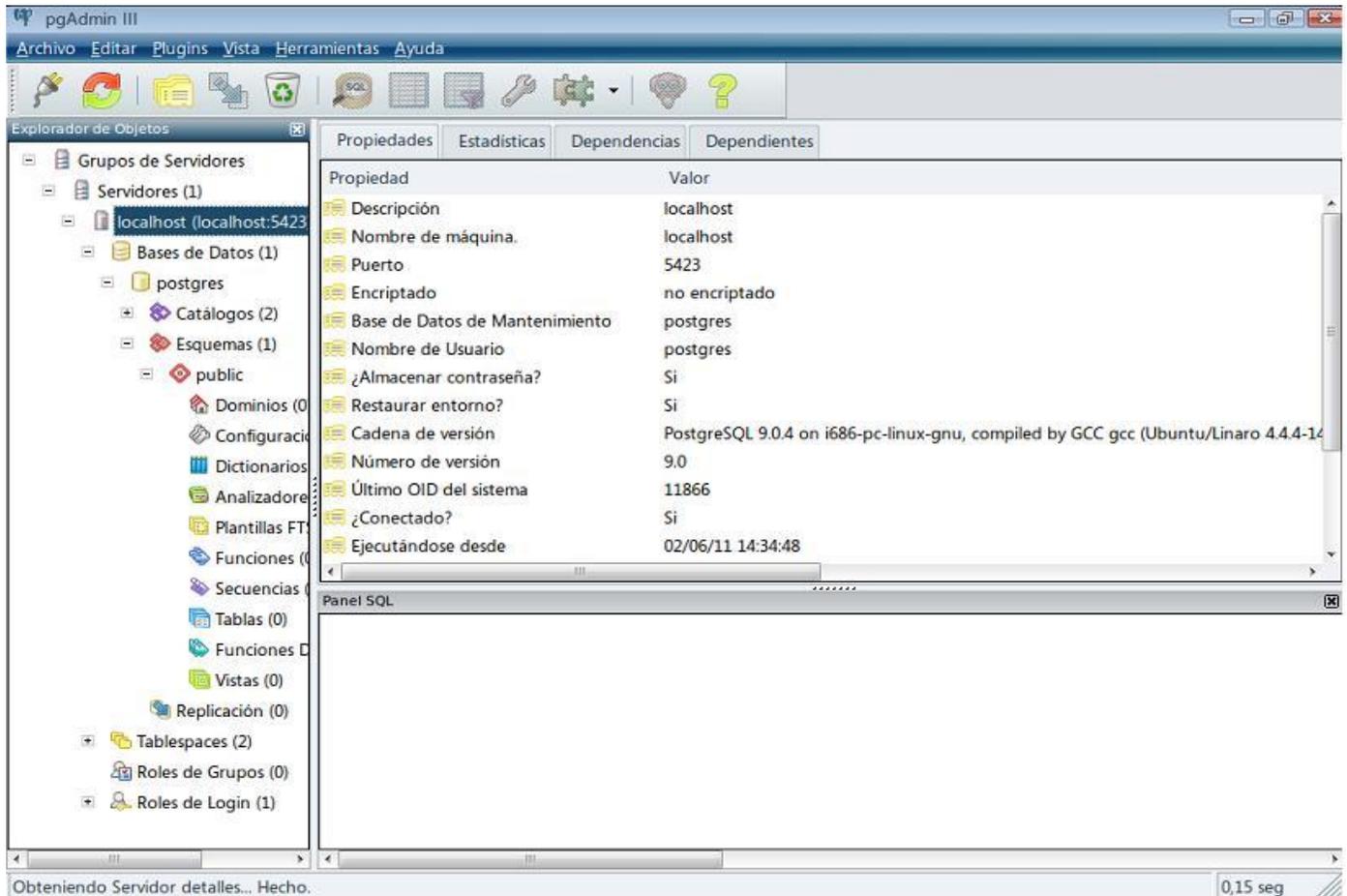


Figura 20: Herramienta pgAdmin3en conexión con el PostgreSQL 9.0.4.

3.4 Funcionamiento de los lenguajes procedurales

Se realizaron pruebas para verificar el funcionamiento de los Lenguajes Procedurales, ejecutando una consulta para cada uno de ellos como son: “*CREATE LANGUAGE plperl*”, “*CREATE LANGUAGE plpythonu*” y “*CREATE LANGUAGE plpgsql*” y fue satisfactoria como se muestra en la Figura 21 que aparece a continuación.

```
CREATE LANGUAGE plperl
La consulta se ejecutó con éxito sin resultado en 183 ms.

CREATE LANGUAGE plpythonu
La consulta se ejecutó con éxito sin resultado en 667 ms.

CREATE LANGUAGE pltcl
La consulta se ejecutó con éxito sin resultado en 233 ms.
```

Figura 21: Lenguajes procedurales creados satisfactoriamente.

3.5 Funcionamiento de los módulos del *contrib*

En la figura que se muestra a continuación se evidencia el funcionamiento correcto de los módulos del *contrib*, para ello fue seleccionado de forma aleatoria una muestra de cuatro módulos y a cada uno les fue ejecutada una consulta para dicha verificación.

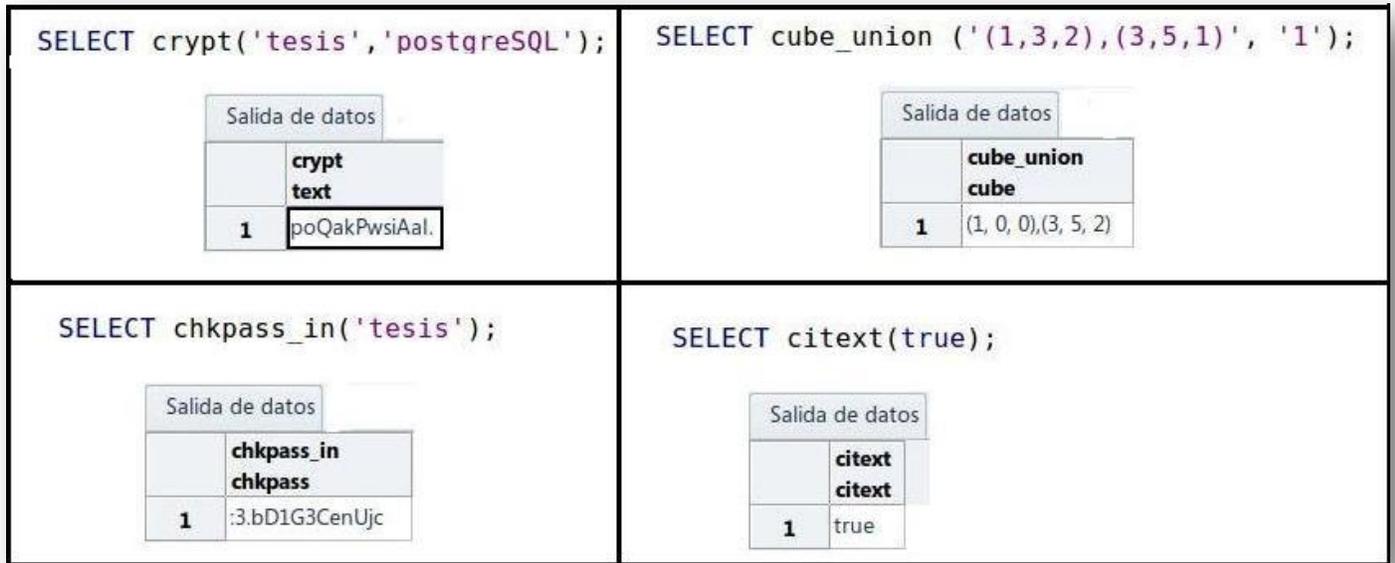


Figura 22: Funcionamiento de los módulos del *contrib*.

Conclusiones

En este capítulo se realizaron las pruebas de caja negra al instalador avanzado de PostgreSQL 9.0.4, se comprobó la creación y funcionamiento de los accesos directos de la herramienta de administración del gestor, así como su documentación y la del PostgreSQL 9.0.4, que brinda el instalador, se demostró el buen funcionamiento del gestor y de los lenguajes procedurales.

Conclusiones Generales

El instalador avanzado de PostgreSQL 9.0.4 con funcionalidades de análisis de datos, monitoreo, administración, desarrollo y seguridad es desarrollado para que empresas y organismos cubanos adopten fácilmente el gestor, evitando que deban compilar el mismo para realizar una instalación avanzada. El instalador permite que los usuarios instalen la versión 9.0.4 de PostgreSQL, seleccionen lenguajes procedurales, definan los parámetros de configuración e instalen aquellos tipos de módulos que necesitan sin necesidad de compilar éstos, haciendo la instalación menos engorrosa y favoreciendo la asimilación del gestor.

El instalador permite en su totalidad:

- Instalar el gestor en su versión más actualizada.
- Instalar 3 lenguajes procedurales.
- Instalar 30 módulos de contribución.
- Instalar opciones del fichero de configuración.
- Instalar la herramienta de administración del gestor (pgAdmin3).
- Crea accesos directos en el escritorio de la documentación del gestor, la documentación de la herramienta de administración pgAdmin3 y la propia herramienta.

Fue probado el funcionamiento correcto del instalador en los sistemas operativos Ubuntu 10.10 y Debian. Por todo lo anterior se concluye que los objetivos propuestos para la presente investigación han sido cumplidos satisfactoriamente.

Referencia bibliográfica

1 PostgreSQL <http://www.postgresql.org>.

2 DesarrolloWeb.com <http://www.desarrolloweb.com/articulos/sistemas-gestores-bases-datos.html>

3 Garbage

Collector http://www.error500.net/garbagecollector/archives/categorias/bases_de_datos/sistema_gestor_de_base_de_datos_sgbid.php

4 Maestros del Web <http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>

5 *Guía para la personalización de PostgreSQL 8.4*. La Habana Tesis de Pregrado 2010

6 <http://www.postgresql.org/about/press/presskit90.html.es>

7 <http://www.ibiblio.org/pub/linux/docs/LuCaS/Postgresql-es/web/navegable/todopostgresql/xplang.html>.

8 <http://www.elmundolinix.com/definicionlinux.php>

9 http://www.programacion.com/noticia/nueva_version_de_izpack_3_3_0_555.

10 <http://izpack.org/documentation/introduction.html>

11 <http://zonalinux.com.ar/ubucompiler-programa-open-source-para-crear-instaladores-deb>.

12 <http://installbuilder.bitrock.com>

13 http://www.guia-ubuntu.org/index.php?title=PgAdmin_III

14 <http://www.buenastareas.com/ensayos/Pruebas-De-Caja-Negra/1477804.htm>

15 <http://iie.fing.edu.uy/~vagonbar/gcc-make/gcc.htm>